Content Sharing in Mobile Networks with Infrastructure: Planning and Management

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO
Dottorato in Ingegneria Elettronica
e delle Comunicazioni – XXIV ciclo

Tesi di Dottorato

Content Sharing in Mobile Networks with Infrastructure

Planning and Management

Francesco MALANDRINO
Matr. 160485

Tutore
prof. Claudio CASETTI

Novembre 2011

Summary

This thesis focuses on mobile ad-hoc networks (with pedestrian or vehicular mobility) having infrastructure support. We deal with the problems of design, deployment and management of such networks.

A first issue to address concerns infrastructure itself: how pervasive should it be in order for the network to operate at the same time efficiently and in a cost-effective manner? How should the units composing it (e.g., access points) be placed? There are several approaches to such questions in literature, and this thesis studies and compares them. Furthermore, in order to effectively design the infrastructure, we need to understand how and how much it will be used. As an example, what is the relationship between infrastructure-to-node and node-to-node communication? How far away, in time and space, do data travel before its destination is reached?

A common assumption made when dealing with such problems is that perfect knowledge about the current and future node mobility is available. In this thesis, we also deal with the problem of assessing the impact that an imperfect, limited knowledge has on network performance.

As far as the management of the network is concerned, this thesis presents a variant of the paradigm known as publish-and-subscribe. With respect to the original paradigm, our goal was to ensure a high probability of finding the requested content, even in presence of selfish, uncooperative nodes, or even nodes whose precise goal is harming the system. Each node is allowed to get from the network an amount of content which corresponds to the amount of content provided to other nodes. Nodes with caching capabilities are assisted in using their cache in order to improve the amount of offered content.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Ad hoc wireless networks have emerged several decades ago as a promising paradigm, with plenty of interesting theoretical aspects and potential practical applications. Over time, only some specific kinds of ad hoc networks have reached enough *critical mass* to be actually deployed. With the partial (yet significant) exception of sensor networks, most ad hoc networks existing today (and, conceivably, most future ones) have *infrastructure* support. As far as applications are concerned, ad hoc networks have proved to be an effective way to assist users that want to *share* some contents, either self-produced or downloaded from the Internet.

Content sharing in mobile networks with infrastructure support is the topic of this thesis. We consider the two distinct, yet strictly linked, aspects of *planning*, i.e., how infrastructure should be deployed, and *management*, i.e., how the network should work.

Deploying the infrastructure means, first and foremost, placing the units composing it (e.g., access points) in a way that maximizes the global performance – i.e., the throughput. This problem is addressed in Chapter 2.

Our solution is able to process any mobility trace, either real or synthetic, without making restrictive hypotheses on connectivity (e.g., on the link between node distance and network rate). Furthermore, we are able to compare the optimal AP placement with the one resulting from other, straightforward and/or popular, heuristics (e.g., placing APs in the most crowded locations).

The key idea of our approach is to describe the positions of mobile nodes and candidate AP locations through a graph. Vertices correspond to mobile nodes and AP locations, and edges describe the connection opportunities among them. Vertices, edges and their properties (e.g., the rate associated to edges) change over time; we are able to capture such changes through a node-splitting technique (i.e., for each node in the physical network, we create several vertices in the graph). Once the graph is built, we can use well-known algorithms and tools to find the AP deployment that guarantees the maximum throughput (i.e., the maximum flow on the graph). Furthermore, it is easy to force a different, suboptimal AP deployment in order to study the maximum flow such

a deployment can yield (and how far from the optimum it is).

In Chapter 3, we further extend our work to take into account the uncertainty affecting the available knowledge of user mobility. When a (potential) relay node passes by the coverage area of an AP, the latter has to decide which data, if any, should be sent to the first. This decision is based on the knowledge (or the forecast) of the downloaders (i.e., nodes interested in actually downloading data) the relay will meet during the remainder of its trip. Wrong or inaccurate predictions can lead to a waste of bandwidth. Our model represents the prediction inaccuracy using only three synthetic parameters, sufficient to distinguish the different existing prediction techniques, and to study their effectiveness.

In Chapter 4, we switch to a non-cooperative scenario, in which the AP deployment is not decided in a centralized way, but is the result of the action of several competing operators. We study the case of a road segment with asymmetric vehicular flows, and two AP locations at its extremes. Counter-intuitively enough, which position is best (in terms of successfully transmitted traffic) depends in a non-trivial way upon the relation between the vehicular flows, as well as the size of the content being transferred.

Chapter 5 deals with the *management* of the network, i.e., how contents are discovered and exchanged among users. Specifically, we present and analyze a content discovery solution, based on a variant of the paradigm known as publish-and-subscribe. Mobile users (Agents) produce and exchange content items, while one or more Brokers, accessible through the infrastructure, join demand and offer, deciding each time which Agent has to provide the requested content. Unlike the basic publish-and-subscribe paradigm, users cannot refuse to provide a content when asked by the Broker.

In order to ensure a prompt content discovery, we 1) make it convenient for rational (i.e., self-interested) users to cooperate when required by the Broker; 2) counter those Agents whose sole purpose is disrupting the system; 3) increase the availability of contents, allowing Agents to use their cache. More exactly, we associate to each Agent a balance, reflecting the difference between the service provided to the network and the service obtained from it. We show, using a game-theoretic approach, that this system yields a Nash equilibrium where all (rational) Agents follow the Broker's indications, and such an equilibrium is also Pareto-optimal. We also define a feedback mechanism, allowing us to identify those Agents (free-riders) that do not provide the announced services. Finally, in order to increase the availability of the least common contents, Brokers can suggest Agents to copy such contents in their cache. They reduce the load on the Agents following their suggestions, thus incentivizing them to do so.

Chapter 6 presents the conclusions we draw from this work.

During the research stay at the University of California, Irvine, a different type of network with infrastructure has been dealt with – cellular networks. These networks are challenged by the phenomenon known as flash-crowd: a large number of users, often geographically close to each other, trying to fetch (almost) contemporarily the same

content, which gained a sudden popularity through social networks (e.g., the *like* button of Facebook).

The spread of interest has been extensively studied in the field of sociology, and there are several models able to forecast, given the "friendship" network, which users are more likely to request a certain content in the next future. Exploiting such a knowledge, network operators can push the content to such users *before* they request it. The global effect is making the bandwidth consumption more regular over time. Since cellular networks are provisioned for the peak (non the average) traffic, major economic saving for operators would follow. The main challenges arising from this *proactive seeding* technique are related to privacy (users may not want to disclose their friendship network) and to prediction models (accurate as they may be, they are still prone to false positives, which may jeopardize the bandwidth saving). Proactive seeding is presented in Appendix A.

Chapter 2

Optimal infrastructure planning

2.1 Introduction

Our case study for infrastructure planning is a vehicular network, where mobile users need to download some content during their trip. Examples of applications of vehicular communication abound, and range from the updating of road maps to the retrieval of nearby points of interest, from the instant learning of traffic conditions to the download of touristic information and media-rich data files.

Within such a context, previous works on content downloading in vehicular networks have dealt with individual aspects of the process, such as the deployment of roadside Access Points (APs) [1–3], the performance evaluation of I2V communication [4], or the exploitation of specific V2V transfer paradigms [5, 6]. None of them, however, has tackled the problem as a whole, trying to quantify the actual potential of an I2V/V2V-based content downloading. In this chapter, we identify the downloading performance limits achievable through DSRC-based I2V/V2V communication.

To this end, we assume ideal conditions from a system engineering viewpoint, i.e., the availability of preemptive knowledge of vehicular trajectories and perfect scheduling of data transmissions, and we cast the downloading process to a mixed integer linear programming (MILP) max-flow problem. The solution of such a problem yields the optimal AP deployment over a given road layout, and the optimal combination of any possible I2V and V2V data transfer paradigm. It thus represents the theoretical upper bound to the downloading throughput, under the aforementioned assumptions.

While it is true that the resulting problem is NP-complete, we show that, with a careful design of the model, it can be solved in presence of realistic vehicle mobility in a real-world road topology. In addition, we propose a sampling-based technique that efficiently yields a solution even for large-scale instances. Although the problem formulation and the performance figures we derive are interesting per-se, we also exploit our optimal solution to discuss the impact of key factors such as AP deployment, transfer

paradigms and technology penetration rate.

As a final remark, we stress that our model, the first of its type to our knowledge, targets the general case of users interested in best-effort downloading of *different* data content. As a consequence, the goal is not to study information dissemination or cooperative caching, but to investigate the performance of content downloading.

The remainder of the chapter is organized as follows. Sec. 2.2 discusses previous work, while Sec. 2.3 describes the network scenario and the objectives of our work. In Sec. 2.4, we build the graph modeling the vehicular network, while we formulate the max-flow problem in Sec. 2.5. There, we also propose a sampling-based technique to deal with large instances of the problem. Results, derived in the scenarios described in Sec. 2.6, are presented in Sec. 2.7. In Sec. 2.8, we evaluate the impact of our assumptions on the physical and MAC layers through ns-3 simulations. Finally, Sec. 2.9 summarizes our major findings and points out directions for future research.

2.2 Related work

Our work relates to infrastructure deployment and content delivery in mobile environments, as well as to delay tolerant networks. Below, we review the studies that are most relevant to ours, highlighting the novelty of our approach.

Infrastructure deployment. Earlier studies [7,8] focus on the feasibility of using IEEE 802.11 APs to inject data into vehicular networks, as well as on the connectivity challenges posed by such an environment. In [9], the authors show that a random distribution of APs over the street layout can help routing data within urban vehicular ad hoc networks. In [10], the impact of several AP deployments on delay-tolerant routing among vehicles is studied. More precisely, each AP is employed as a static cache for content items that have to be transferred between vehicles visiting the AP at different times. Other than in the scope, the works in [9, 10] differ from ours also because they do not provide theoretical justification of the AP placements they propose.

AP deployment is formulated as an optimization problem in [11, 12], where, however, the objective is not content downloading but the dissemination of information to vehicles in the shortest possible time. The study in [13], instead, estimates the minimum number of infrastructure nodes to be deployed along a straight road segment so as to provide delay guarantees to the data traffic that vehicles have to deliver to the infrastructure, possibly with the help of relays. A similar problem is addressed in [14], with the aim to support information dissemination. The different objectives of the above studies lead to completely different formulations, thus to results not comparable with the ones we present.

In [1, 2], infrastructure placement strategies are proposed that maximize the amount of time a vehicle is within radio range of an AP. Although longer periods of time under coverage can undoubtedly favor the download of contents by vehicular users, important

differences with our work exist. First, our analysis is not limited to direct transfers from APs to vehicles, but includes traffic relaying. Second, while the problem formulation in [1] guarantees a minimum coverage requirement and the one in [2] maximizes the minimum-contact opportunity, we optimize the actual throughput, accounting for the airtime conflicts deriving from the contemporary presence of an arbitrary number of vehicles. Third, instead of studying a predefined set of paths over a given topology we process complete mobility traces.

An AP deployment strategy designed to favor content download through relaying in vehicular networks is introduced in [3]. The proposed optimization problem, however, aims at maximizing a metric reflecting the amount of vehicular traffic that enables V2V communication, and not the actual throughput. Moreover, such a formulation cannot capture the mutual interference among concurrent traffic transfers.

Content downloading and dissemination. With regard to content downloading in vehicular networks, the study in [15], unlike ours, focuses on the access to Web search and presents a system that makes such a service highly efficient by exploiting prefetching. Experimental and analytical results show the contribution of V2V and I2V communications to the system performance. The works in [5, 16] address the benefits of prefetching jointly with traffic scheduling techniques. In particular, the objective of [16] is to maximize the amount of data downloaded by vehicles through APs that form a wireless mesh network, given the AP deployment and an (imprecise) knowledge of the vehicles trajectory and of their connectivity with the APs. However, no multihop data transfer are investigated. In [5], both I2V and V2V communications are considered and the performance evaluation is carried out through simulation and a testbed on a circular campus bus route. Furthermore, a comparison against the solution to a max-flow problem is presented, but (i) it is limited to a simplified, highway-like scenario featuring one AP and one downloader and (ii) it assumes atomic contacts between network nodes, hence neglecting interference and channel contention.

Our study also relates to cooperative downloading in vehicular networks. In this context, the work in [17, 18] introduces a vehicular peer-to-peer file sharing protocol, which allows vehicles to share a content of common interest. Our study on content download, instead, works in the more generic case where each user can be interested in a different file. System assumptions similar to the ones made in [17, 18] are behind the works in [19, 20], about which, as a consequence, the same considerations hold.

Delay tolerant networks (DTNs). The vehicular cooperation paradigm that we consider relates our work to DTNs. In particular, in [21] both experimental results from a real testbed and an asymptotic analysis are carried out to assess the benefit to content dissemination of adding varying numbers of base stations, mesh nodes and relay nodes to a DTN.

A DTN time-invariant graph, which is similar to the time-expanded graph used in

our study, was presented in [22]. With respect to this work, we do not assume the contacts between mobile nodes to be atomic but allow them to have arbitrary duration, and we build the network graph so as to account for the presence of roadside infrastructure and channel contention. The representation of a time-varying network topology as a time-expanded graph can be found in [23, 24], where the former is an earlier version of this work. As for the latter, such a representation is used to identify the nodes whose limited storage may impair the network performance, and to formulate a max-flow problem whose solution leads to an optimal, distributed routing and storage policy. In our work, we address the performance limits of content downloading and the problem of AP deployment, for which no distributed solution is needed.

2.3 Network System and Goals

We envision a network composed of fixed roadside APs and vehicular users, where some of the latter (hereinafter named *downloaders*) are interested in downloading best-effort traffic from the Internet through the APs. We consider the general case in which every downloader may be interested in different content: downloaders can either exploit direct connectivity with the APs, if available, or be assisted by other vehicles acting as intermediate relays. Specifically, we account for all possible data *transfer paradigms* that can be implemented through I2V/V2V communication:

- **direct transfer**, resulting from a direct communication between an AP and a downloader. This represents the typical way mobile users interact with the infrastructure in today's wireless networks;
- **connected forwarding**, i.e., traffic relaying through one or more vehicles that create a multi-hop path between an AP and a downloader, where all the links of the connected path exist at the time of the transfer. This is the traditional approach to traffic delivery in ad hoc networks;
- **carry-and-forward**, i.e., traffic relaying through one or more vehicles that store and carry the data, eventually delivering them either to the target downloader or to another relay deemed to meet the downloader sooner.

We stress that connected forwarding and carry-and-forward are inherently multi-hop paradigms. We assume that vehicular users are rational, hence they can be engaged in relaying traffic for others only if they are not currently retrieving the content for themselves. Furthermore, since our goal is to derive an upper bound to the system performance, we assume the availability of preemptive knowledge of vehicular trajectories and perfect scheduling of data transmissions.

From the viewpoint of the network system, we consider that each node (a vehicle or an AP) has one radio interface only. This is a common assumption for vehicular nodes,

while the extension to the case where APs have more than one interface is straightforward. Any two nodes in the network can communicate at a given time instant, i.e., they are neighbors, if their distance is below or equal to their maximum radio range. Also, we assume that the maximum radio range is common to all network nodes and is equal to the node interference range[1]. We consider that V2V communication occurs on the same frequency channel, which is different from the channels used for I2V communication[2]. APs with overlapping coverage areas operate on separate channels as well. When under AP coverage, a vehicle can always choose either I2V or V2V communication. The nodes share the channel bandwidth allocated for service applications using an IEEE 802.11-based MAC protocol.

Our objective is to design the content downloading system so as to maximize the aggregate throughput. To this aim, we have to jointly solve two problems: (i) given a set of candidate locations and a number of APs to be activated, we need to identify the deployment yielding the maximum throughput; (ii) given the availability of different data transfer paradigms, possibly involving relays, we have to determine how to use them in order to maximize the data flow from the infrastructure to the downloaders. Our approach consists in processing a road layout and an associated vehicular mobility trace, so as to build a graph that represents the temporal network evolution (Sec. 2.4). By using this graph, we formulate a max-flow problem whose solution matches our goals (Sec. 2.5).

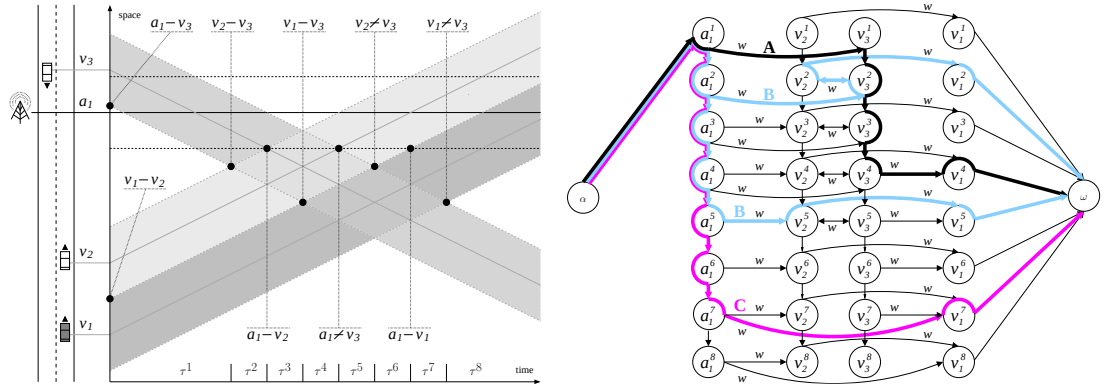
2.4 Dynamic Network Topology Graph

We generate a time-expanded graph [25], hereinafter dynamic network topology graph (DNTG), from a vehicular mobility trace. To build the graph, we consider that on the road layout corresponding to the mobility trace there are: (i) a set of A candidate locations (a_i, $i = 1,\ldots,A$) where APs could be placed, (ii) a set of V vehicles (v_i, $i = 1,\ldots,V$) transiting over the road layout and participating in the network, and (iii) a subset of D vehicles that wish to download data from the infrastructure.

The aim of the DNTG is to model all possible opportunities through which data can flow from the APs to the downloaders, possibly via relays. Given the mobility trace, we therefore identify the *contact events* between any pair of nodes (i.e., two vehicles, or an AP and a vehicle). Each contact event is characterized by:
(i) the quality level of the link between the two nodes. Several metrics could be considered; here, we specifically take as link quality metric the data rate achievable at the network layer;

[1] Although simplistic, the impact of such an assumption on the system performance is negligible, as shown by the comparison between analytical and simulation results in Sec. 2.8.

[2] Single-radio multichannel management is foreseen by current standardization activities on vehicular communication systems, e.g., IEEE 1609.4.



(a) Space-time representation of the contact events (b) DNTG resulting from the contact events

Figure 2.1. A sample set of contact events (a) and the corresponding DNTG (b), in presence of one candidate AP location and three vehicles, the first of which (v_1) is a downloader while the others (v_2,v_3) can act as relays. In (a), shadowed areas represent *halved* transmission ranges, so that links exist when two shadowed areas touch or overlap, and break when such areas become disjoint. These events allow to fragment the time into frames of duration $\tau^1,\ldots,\tau^8$ (for simplicity, here the link quality is assumed constant). The network connectivity during each frame is represented by a row of vertices in the DNTG. In the graph, we highlight paths that are representative of the carry-and-forward (A), connected forwarding (B), and direct (C) transfer paradigms

(ii) the contact starting time, i.e., the time instant at which the link between the two nodes is established or the quality level of an already established link takes on a new value;
(iii) the contact ending time, i.e., the time instant at which the link is removed, or its quality level has changed.
We stress that, by associating a time duration to the contact events, instead of considering them as atomic, we can model critical aspects of real-world communication, such as channel contention.

The time interval between any two successive contact events in the network is called *frame*. Within a frame the network is static, i.e., no link is created or removed and the link quality levels do not change. We denote by F the number of frames in the considered trace, and by τ^k the duration of the generic frame k ($1 \leq k \leq F$); also, all on-going contact events during frame k are said to be *active* in that frame.

Each vehicle v_i participating in the network at frame k is represented by a vertex v_i^k ($1 \leq i \leq V$) in the DNTG, whereas every candidate AP location a_i is mapped within each frame k onto a vertex a_i^k ($1 \leq i \leq A$). We denote by $\mathcal{V}^k$ and $\mathcal{A}^k$ the set of vertices representing, respectively, the vehicles and candidate AP locations in the DNTG at time frame k, while we denote by $\mathcal{D}^k \subseteq \mathcal{V}^k$ the subset of vertices representing the downloaders that exist in the network at frame k. All non-downloader vehicles in

$\mathcal{R}^k = \mathcal{V}^k \setminus \mathcal{D}^k$ can act as relays, according to the data transfer paradigms outlined above.

Within each frame k, a directed edge (v_i^k,v_j^k) exists from vertex $v_i^k \in \mathcal{R}^k$ to vertex $v_j^k \in \mathcal{V}^k$, if a contact between the non-downloader vehicle v_i and another vehicle v_j is active during that frame. Each edge of this type is associated with a weight $w(v_i^k,v_j^k)$, equal to the rate of that contact event. The set including such edges is defined as $\mathcal{L}_v^k$. Similarly, a directed edge (a_i^k,v_j^k) exists from vertex $a_i^k \in \mathcal{A}^k$ to vertex $v_j^k \in \mathcal{V}^k$, if a contact between the candidate AP location a_i and the vehicle v_j is active during frame k. Again, these edges are associated with weights $w(a_i^k,v_j^k)$, corresponding to the contact event rate, and their set is defined as $\mathcal{L}_a^k$.

A directed edge (v_i^k,v_i^{k+1}) is also drawn from any vertex $v_i^k \in \mathcal{R}^k$ to any vertex $v_i^{k+1} \in \mathcal{R}^{k+1}$, for $1 \leq k < F$. While the edges in $\mathcal{L}_v^k$ and $\mathcal{L}_a^k$ represent transmission opportunities, those of the form (v_i^k,v_i^{k+1}) model the possibility that a non-downloader vehicle v_i physically carries some data during its movement in the time interval from frame k to frame $k+1$. Accordingly, these edges are associated with a weight representing the vehicle memory capabilities, since they do not imply any rate-limited data transfer over the wireless medium. However, dealing with vehicular nodes as opposed to resource-constrained hand-held devices, we assume the weight of such edges to take on an infinite value. A directed edge (a_i^k,a_i^{k+1}) of infinite weight is also drawn between any two vertices representing the same candidate AP location at two consecutive frames, i.e., from $a_i^k \in \mathcal{A}^k$ to $a_i^{k+1} \in \mathcal{A}^{k+1}$ ($1 \leq k < F$). We will refer to the edges of the kind (v_i^k,v_i^{k+1}) or (a_i^k,a_i^{k+1}) as intra-nodal.

Finally, in order to formulate a max-flow problem over the DNTG, we introduce two virtual vertices, α and ω, respectively representing the source and destination of the total flow over the graph. Then, the graph is completed with infinite-weight edges (α,a_i^1), from α to any vertex $a_i^1 \in \mathcal{A}^1$, and (v_i^k,ω), from any vertex $v_i^k \in \mathcal{D}^k$ to ω, $1 \leq k \leq F$.

The DNTG is therefore a weighted directed graph, representing the temporal evolution of the network topology. An example of its derivation is given in Fig. 2.1, in presence of one AP location and three vehicles v_1,v_2,v_3, with v_1 being a downloader and v_2,v_3 possibly acting as relays. Fig. 2.1(a) depicts the spatio-temporal evolution of node positions: there, contact events are highlighted through the times at which links are established or lost. For simplicity, in this example we assume the achievable network-layer rate w to be constant during the entire lifetime of a link. The durations of the frames, within which the network connectivity is unchanged, are denoted by $\tau^1,\ldots,\tau^8$. In Fig. 2.1(b), frames correspond to rows of vertices in the DNTG, where intra-nodal edges connect vertices representing the same vehicle or candidate AP location over time. Note that the graph allows us to capture all the data transfer paradigms previously discussed. It is thus possible to identify paths in the graph that correspond to (i) direct download from the candidate AP to the downloader, as path C, (ii) connected forwarding through 3-hops (frame 2) and 2-hops (frame 5), as path B, and (iii) carry-and-forward

through the movement in time of the relay v_3, as path A.

2.5 The Max-Flow Problem

Given the DNTG, our next step is the formulation of an optimization problem whose goal is to maximize the flow from α to ω, i.e., the total amount of downloaded data. Denoting by $x(\cdot,\cdot)$ the traffic flow over an edge connecting two generic vertices, our objective can be expressed as:

$$\max \sum_{k=1}^{F} \sum_{v_i^k \in \mathcal{D}^k} x(v_i^k,\omega)\,. \tag{2.1}$$

The max-flow problem needs to be solved taking into account several constraints, listed below.

2.5.1 Constraints

Non-negative flow. The flow on every existing edge must be greater than or equal to zero.

Flow conservation. For any vertex in the DNTG, the amount of incoming flow must equal the amount of outgoing flow. This constraint is expressed in slightly different form, depending on whether the vertex represents a downloader, a relay, or a candidate AP location. For the generic vertex representing a downloader, $v_i^k \in \mathcal{D}^k$, and any frame k, this maps onto:

$$\sum_{\substack{a_j^k \in \mathcal{A}^k: \\ (a_j^k,v_i^k)\in\mathcal{L}_a^k}} x(a_j^k,v_i^k) + \sum_{\substack{v_j^k \in \mathcal{R}^k: \\ (v_j^k,v_i^k)\in\mathcal{L}_v^k}} x(v_j^k,v_i^k) = x(v_i^k,\omega)\,. \tag{2.2}$$

For any frame k and potential relay vertex, $v_i^k \in \mathcal{R}^k$:

$$\begin{aligned} &\sum_{\substack{a_j^k \in \mathcal{A}^k: \\ (a_j^k,v_i^k)\in\mathcal{L}_a^k}} x(a_j^k,v_i^k) + \sum_{\substack{v_j^k \in \mathcal{R}^k: \\ (v_j^k,v_i^k)\in\mathcal{L}_v^k}} x(v_j^k,v_i^k) + \mathbb{1}_{\left[(v_i^{k-1},v_i^k)\right]} x(v_i^{k-1},v_i^k) \\ &= \sum_{\substack{v_j^k \in \mathcal{V}^k: \\ (v_i^k,v_j^k)\in\mathcal{L}_v^k}} x(v_i^k,v_j^k) + \mathbb{1}_{\left[(v_i^k,v_i^{k+1})\right]} x(v_i^k,v_i^{k+1}) \end{aligned} \tag{2.3}$$

where the indicator function is equal to 1 if the specified edge exists, and it is 0 otherwise.

For each vertex representing a candidate AP, $a_i^k \in \mathcal{A}^k$:

$$\mathbb{1}_{[k=1]} x(\alpha,a_i^k) + \mathbb{1}_{[k>1]} x(a_i^{k-1},a_i^k) = \mathbb{1}_{[k<F]} x(a_i^k,a_i^{k+1}) + \sum_{\substack{v_j^k \in \mathcal{V}^k: \\ (a_i^k,v_j^k)\in\mathcal{L}_a^k}} x(a_i^k,v_j^k) \tag{2.4}$$

where the indicator functions are equal to 1 if the specified condition holds, and 0 otherwise. Note that, for a vertex representing a candidate AP location, ingoing flows may come from a vertical edge or from α, while outgoing flows may be over a vertical edge, or over edges toward vehicle vertices.

Finally, we impose that the total flow exiting α equals the total flow entering ω: $\sum_{a_i^1 \in \mathcal{A}^1} x(\alpha,a_i^1) = \sum_{k=1}^{F} \sum_{v_j^k \in \mathcal{D}^k} x(v_j^k,\omega)$.

Channel access. As mentioned, we deal with unicast transmissions and assume that the nodes use a 802.11-based MAC scheme; also, V2V and I2V communications occur on different channels. Then, given a tagged vehicle, we consider that none of the following events can take place simultaneously, and the time span of each frame must be shared among them:

1. the vehicle transmits to a neighboring vehicle;
2. a neighboring vehicle receives from any relay;
3. the vehicle receives from a neighboring relay;
4. a neighboring relay transmits to any vehicle;
5. the vehicle receives from a neighboring AP.

Recall that we do not model the scheduling of the single packets transmitted within each frame. Rather, we consider the total amount of data carried by each flow. Also, in 2) a neighboring vehicle receiving data is accounted for; in presence of hidden terminals, this still holds if the RTS/CTS handshake is used. Considering that: 1) is a subcase of 2) and 3) is a subcase of 4), for the generic vertex $v_i^k \in \mathcal{V}^k$ and for any frame k, we have:

$$\sum_{\substack{v_j^k \in \mathcal{R}^k, v_m^k \in \mathcal{V}^k \\ (v_j^k,v_m^k)\in\mathcal{L}_v^k}} \mathbb{1}_{\left[(v_m^k,v_i^k)||(v_j^k,v_i^k)\right]} \frac{x(v_j^k,v_m^k)}{w(v_j^k,v_m^k)} + \sum_{\substack{a_j^k \in \mathcal{A}^k: \\ (a_j^k,v_i^k)\in\mathcal{L}_a^k}} \frac{x(a_j^k,v_i^k)}{w(a_j^k,v_i^k)} \leq \tau^k \tag{2.5}$$

where the indicator functions are defined as before.

In addition, for each candidate AP, we have that its total transmission time during the generic frame k cannot exceed the frame duration. Thus, for any k and $a_j^k \in \mathcal{A}^k$, we

have:

$$\sum_{\substack{v_i^k \in \mathcal{V}^k: \\ (a_j^k, v_i^k) \in \mathcal{L}_a^k}} \frac{x(a_j^k, v_i^k)}{w(a_j^k, v_i^k)} \leq \tau^k \,. \tag{2.6}$$

The previous constraints allow a vehicle under coverage of an AP to use I2V and V2V communications within the same frame. Next, we consider the case where a vehicle under the coverage of (at least) one AP is not configured to operate in ad hoc mode, i.e., it cannot communicate with other vehicles. Then, for any frame k and $v_j^k \in \mathcal{R}^k, v_m^k \in \mathcal{V}^k$ such that $(v_j^k, v_m^k) \in \mathcal{L}_v^k$, the following constraint holds:

$$x(v_j^k, v_m^k) \leq \left(1 - \max_{\substack{a_i^k \in \mathcal{A}^k: \\ (a_i^k, v_j^k) \| (a_i^k, v_m^k) \in \mathcal{L}_a^k}} \{y_i\} \right) w(v_j^k, v_m^k) \tau^k \tag{2.7}$$

where y_i, $i = 1, \ldots, A$, are Boolean variables, whose value is 1 if an AP is placed at candidate location i and 0 otherwise. If there is at least an AP within the vehicle range, the first term of the product becomes 0, thus imposing that the flow on all edges $(v_j^k, v_m^k) \in \mathcal{L}^k$ is 0.

Overlapping AP coverages. Recall that, when a vehicle falls within coverage of two or more APs, we assume that, during a frame, it communicates with one AP only, and that the APs operate on different frequency channels. We therefore introduce a second set of Boolean variables t_{ij}^k ($1 \leq i \leq A$, $1 \leq j \leq V$, $1 \leq k \leq F$) whose value is 1 if the candidate AP a_i communicates with the vehicle v_j during frame k and 0 otherwise. Then, for every candidate AP vertex $a_i^k \in \mathcal{A}^k$, vehicle vertex $v_j^k \in \mathcal{V}^k$, and frame k, we impose that

$$t_{ij}^k \in \{0,1\} \;;\; \sum_{i=1}^{A} t_{ij}^k \leq 1 \;;\; x(a_i^k, v_j^k) \leq w(a_i^k, v_j^k) \tau^k t_{ij}^k \,.$$

Maximum number of APs. The last set of constraints imposes that no more than $\hat{A}$ candidate AP locations are selected, through the variables y_i. Then, for any i, we write:

$$y_i \in \{0,1\} \;;\; \sum_{i=1}^{A} y_i \leq \hat{A} \;;\; x(\alpha, a_i^1) \leq M y_i$$

where $M \in \mathbb{R}$ is an arbitrarily large positive constant.

2.5.2 Modeling the transfer paradigms

We now describe how the different transfer paradigms introduced in Sec. 2.3 are modeled in our formulation.

The traffic transferred through the direct paradigm corresponds to the amount of data that, at any frame k, flows from one candidate AP vertex, $a_i^k \in \mathcal{A}^k$, to $v_j^k \in \mathcal{D}^k$. As for the traffic transferred through connected forwarding, this is represented as the amount of data that, at any frame k, flows from one relay vertex $v_i^k \in \mathcal{R}^k$ to a downloader vertex, with one or more edges connecting v_i^k to the vertex representing the AP that originated the data. Such a situation indeed corresponds to the case where a multi-hop connected path between an AP and a downloader exists. The data transferred through carry-and-forward, instead, correspond to the flow associated with any (v_i^k,v_j^k) edge at frame k (with $v_i^k \in \mathcal{R}^k$,$v_j^k \in \mathcal{D}^k$), such that the relay vertex v_i^k is no longer connected (either directly or through multiple edges) to the vertex representing the AP that originated the flow.

Furthermore, while deriving the results, we consider three possible cases. In the *unlimited* case, no limitation is imposed to the maximum number of relays used to deliver traffic to a downloader. This is modeled simply using the constraints listed in Sec. 2.5.1. In the *2-hop limit* case, at most one relay can be employed. This is studied by imposing that transmissions between relays cannot occur, i.e., $x(v_i^k,v_j^k) = 0$ for $1 \leq k \leq F$ and v_i^k,$v_j^k \in \mathcal{R}^k$, such that $(v_i^k,v_j^k) \in \mathcal{L}_v^k$. In the *1-hop limit* case, only 1-hop transfers from an AP to a downloader are allowed; we represent this case by imposing that[3]: $x(v_i^k,v_j^k) = 0$

for any k and $v_i^k \in \mathcal{R}^k$,$v_j^k \in \mathcal{V}^k$ such that $(v_i^k,v_j^k) \in \mathcal{L}_v^k$.

2.5.3 Sampling-based solution

The problem falls in the category of mixed integer linear programming (MILP) problems. We solve the problem through the Gurobi solver, which uses a variant of the branch-and-cut algorithm.

However, due to the large number of constraints involving Boolean variables, solving the MILP on the full DNTG is impractical for large instances (e.g., large geographical areas, high number of vehicles participating in the content downloading, or large number of candidate AP locations). To be able to analyze such cases, we resort to a graph sampling approach. More specifically, we take the following steps:

1) we sample the DNTG obtaining a small, yet representative, sub-graph, which includes all relevant candidate AP locations (as detailed below);

2) we find the optimal AP deployment using such a sub-graph;

[3]The flow conservation constraints ensure that no positive flow exists from a candidate AP vertex to a relay vehicle vertex.

3) we apply the obtained deployment to the full graph and optimize the flows, a linear programming (LP) problem that can be easily solved as it does not involve Boolean variables.

In order to accomplish the first step, the selected sub-graph must include the α and ω vertices, and reflect the characteristics of the original graph (e.g., the relevance of the candidate AP locations). Since we have to collect not only a representative set of the graph vertices, but a *connected* sample, uniform vertex sampling is not a viable option. Thus, we resort to a random walk-based approach [26], and devise a tailored variant of it. Such a variant is needed to effectively cope with the following challenging peculiarities that our DNTG exhibits with respect to ordinary graphs.

First, not only is the DNTG directed, but the flow goes from α to ω, while the edges are specifically directed from candidate AP location vertices to vehicle vertices, as well as from lower to higher values of the frame index k. This implies that it is not possible to make an arbitrarily long walk on the DNTG; thus, we need to combine vertices and edges that are sampled over subsequent multiple short walks.

Second, while walking from α to ω would be a natural choice in ordinary graphs, in our case this would turn into sampling relay vehicle vertices that may not be connected with downloaders, hence with ω (see top Fig. 2.2). To avoid the unnecessary sampling of these vertices, we let the walks go from ω to α, crossing each edge along its opposite direction. Each walk therefore goes from ω to one or more vertices representing one downloader, then possibly to relay vertices, to one candidate AP location and finally to α (see bottom Fig. 2.2). An example is shown in Fig. 2.2(d). Note that, by adopting such a strategy, we obtain a fairly small subgraph, yet containing vertices representing several vehicles (including relays and downloaders), as well as candidate AP locations.

Another desirable effect of the above strategy is that the candidate AP locations and the relays are sampled with a frequency that is proportional to the number of paths between α and ω passing through them, while the downloaders are sampled with a frequency that is proportional to their trip duration. We support such a statement by looking at the correlation between the relevance of candidate AP locations for content downloading, and the number of walks including each candidate AP location. The relevance is expressed as the amount of data per second outgoing from each candidate AP location under the max-flow solution in the full DNTG, in the scenario in Fig. 2.4(a) with $\hat{A} = 60$. As is clear from Fig. 2.3(b), there is a strong correlation when walks start from ω; conversely, with the standard sampling (i.e., for walks starting from α) there is no evident correlation (Fig. 2.3(a)).

The performance obtained by solving the max-flow over the sampled DNTG is compared to that attained by using the full graph in Sec. 2.7.1.

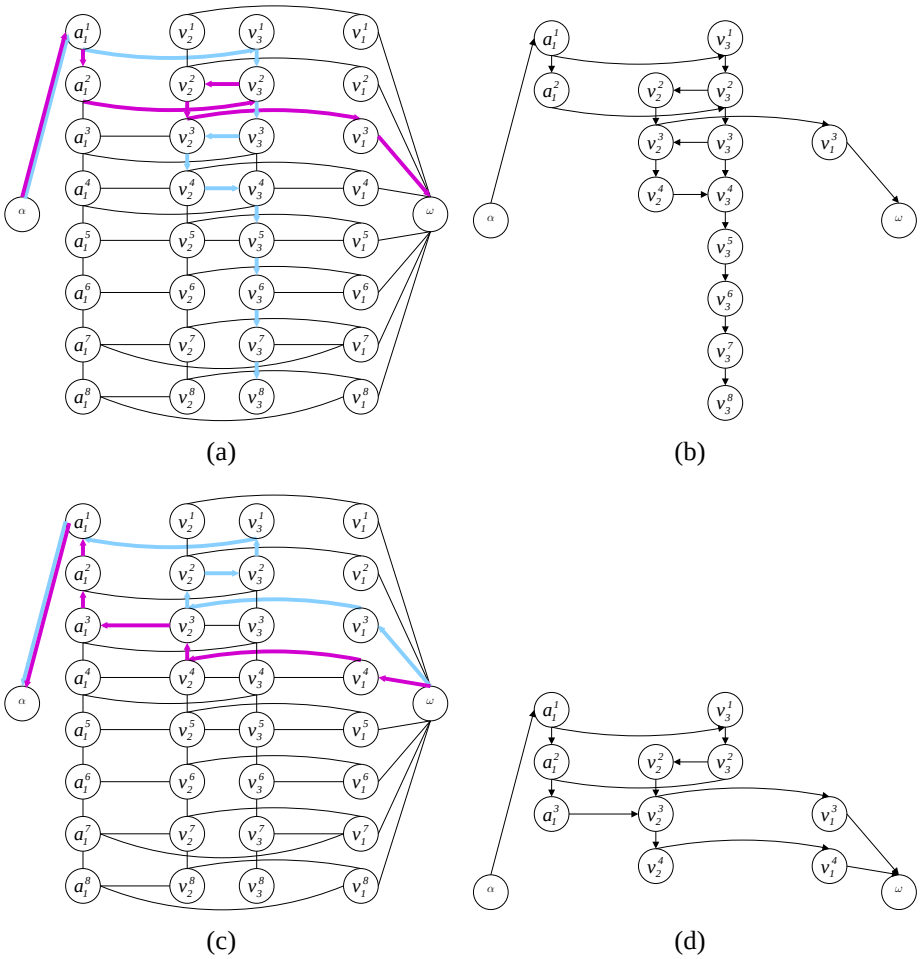
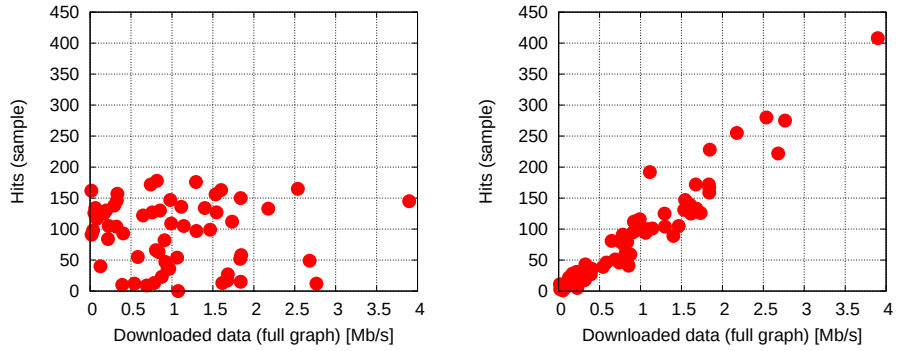


Figure 2.2. Example of a DNTG sampling when walks cross each edge along its direction (a) or its opposite direction (c). Resulting sampled graph are in (b) e (d), respectively. Arrows refer to the walk direction (left figures) and to the edge direction (right figures)

2.6 Reference scenarios

We consider real-world road topologies representing different environments, namely the urban area of Zurich, the village area of Schlieren and the suburban area of Wallisellen, in Switzerland. Each road topology covers an area of 20 km^2; the vehicular mobility in the region has been synthetically generated at ETH Zurich [27]. The macroscopic- and microscopic-level models employed to produce the movement traces allow a realistic representation of the vehicular mobility, in terms of both large-scale traffic flows and small-scale V2V interactions. Although our model can accommodate any frame duration, so as to reflect, e.g., faster variations of the link quality, given the 1-second time granularity of the trace, we consider $\tau^k \geq 1$ s ($k = 1, \ldots, F$).



(a) Random walks from α (b) Random walks from ω

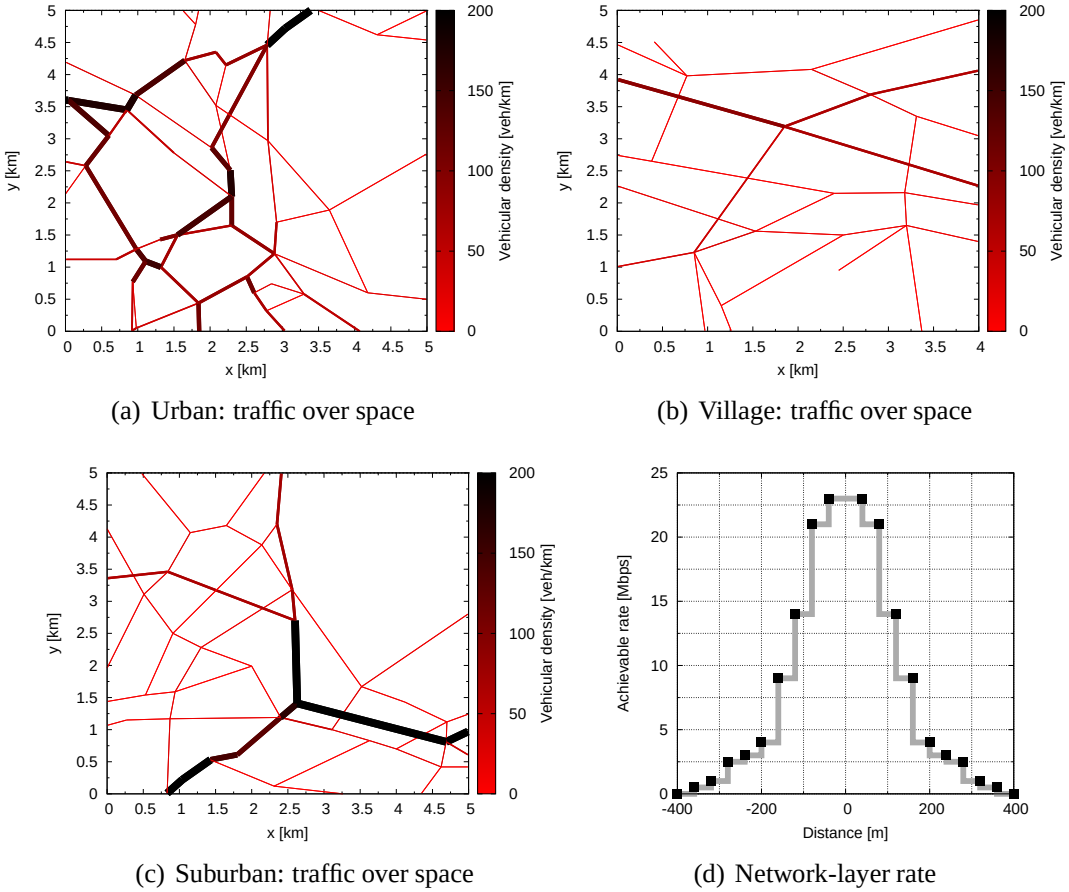
Figure 2.3. Number of times that a vertex corresponding to a candidate AP location is sampled, vs. the amount of data per second downloaded through that AP in the full DNTG. Walks start from α in (a) and from ω in (b)

Since we use a realistic mobility model, in each road topology the vehicular traffic intensity varies depending on the road segment and time of the day. In Figs. 2.4(a)–2.4(c), we report the road layout of the urban, village and suburban environments, highlighting the different traffic volumes observed over each road segment: thicker, darker segments identify the roads characterized by higher vehicular density. As far as vehicular traffic variations over time are concerned, we consider only time periods corresponding to medium-high vehicle density. In the urban, village and suburban traces, each lasting about 5 hours, this leads to an average density of 90, 62.5 and 33.5 veh/km, respectively.

We consider different values of the technology penetration rate, i.e., the fraction of vehicles equipped with a communication interface and willing to participate in the content downloading process; we denote such a parameter by p. Also, the percentage of such communication-enabled vehicles that concurrently request content, i.e., that act as downloaders, is denoted by d. Unless otherwise specified, we will consider $d = 0.01$ (i.e., 1% of the vehicles participating in the network) – a reasonable value as observed in wired networks [28].

The value of the achievable network-layer rate between any two nodes is adjusted according to the distance between them. To this end, we refer to the 802.11a experimental results in [4] to derive the values shown in Fig. 2.4(d), and we use them as samples of the achievable network-layer rate. Note that we limit the maximum node transmission range to 200 m, since, as stated in [4], this distance allows the establishment of a reliable communication in 80% of the cases.

Given the above settings and that $\hat{A}$ APs have to be deployed, in the next section we present the performance obtained by solving the max-flow problem on the full DNTG,



(a) Urban: traffic over space

(b) Village: traffic over space

(c) Suburban: traffic over space

(d) Network-layer rate

Figure 2.4. Road layout in the urban (a), village (b) and suburban (c) scenarios, and characterization of the achievable network-layer rate as a function of distance, based on experimental data (d)

or on its sampled version. We thus attain the optimal AP deployment as well as the values of the flow variables corresponding to the amount of data that downloaders receive. Using the flow values, we can then compute: (i) the per-user throughput, as the ratio of the amount of received data to the downloader trip duration; (ii) the fraction of traffic delivered through the direct, connected forwarding, or carry-and-forward paradigm; (iii) the Jain's fairness index, computed on the average throughput obtained by each downloader; (iv) the average packet delivery delay from AP to downloader, accounting for both I2V and V2V communication.

Our problem formulation can also accommodate any specific AP deployment by fixing the values of the binary variables y_i. The system throughput is then obtained as the output of the max-flow problem given the selected AP locations y_i. Solving the

max-flow problem implies the optimization of the traffic scheduling, i.e., the values taken by the V2V and I2V flow variables at any frame k ($x(v_j^k,v_i^k)$, $x(a_j^k,v_i^k)$). The results thus represent the best performance achievable under the chosen deployment and the assumptions made in Sec. 2.3.

We leverage the capability of our framework to model different AP deployments and explore the following strategies:
Random: $\hat{A}$ locations are randomly selected among the candidate ones, according to a uniform distribution;
Crowded: $\hat{A}$ locations are picked, whose coverage area exhibits, over time, the highest vehicular density;
Contact: $\hat{A}$ locations are selected, which maximize the sum of the contact opportunities between vehicles and APs. Inspired by the metric adopted in [2], for each vehicle we express the contact opportunity as the fraction of the road segment lengths traveled while under coverage of at least one AP.

2.7 Performance evaluation

In this section, we evaluate the performance of content downloading in vehicular networks, by assessing the impact of different settings on the system.

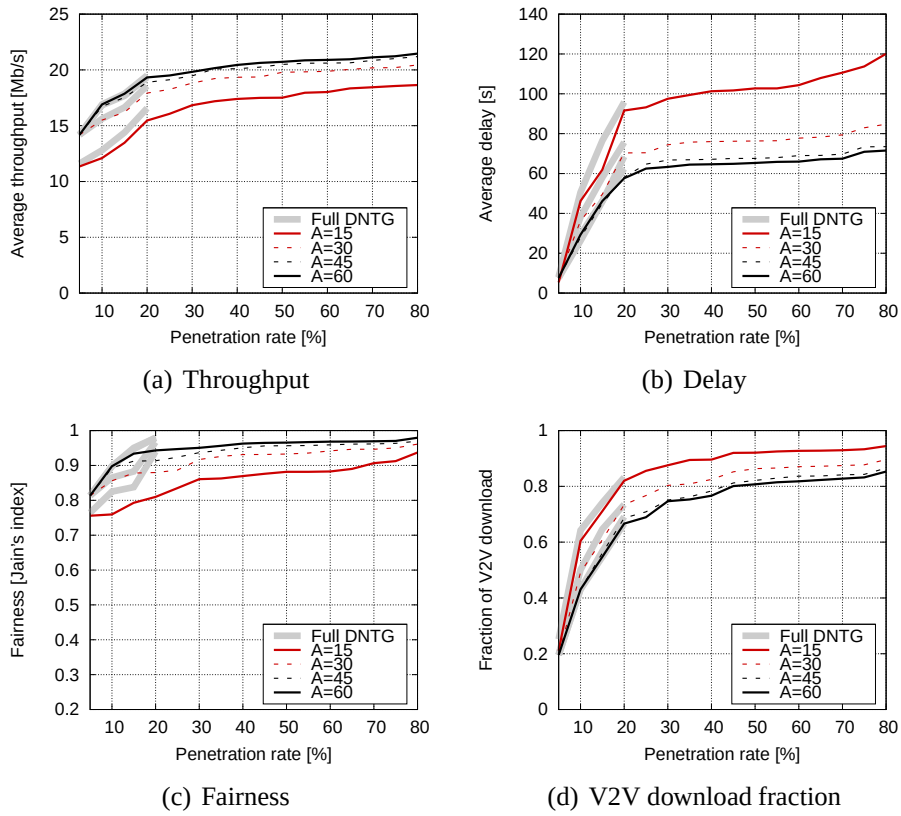
Specifically, in Sec. 2.7.1, we evaluate the impact that the penetration rate of the vehicular communication technology, p, has on the content downloading performance. Our results reveal the existence of two regimes, separating initial deployment stages (characterized by $p < 20\%$) from a mature technology (i.e., $p > 30\%$).

These two working regimes are analysed in detail in Sec. 2.7.2 and Sec. 2.7.3, respectively. For each regime, we discuss the impact of the AP deployment strategies, transfer paradigms and the road environment on the downloading performance. For the high-penetration regime, we also investigate the system behavior as the percentage of concurrent downloaders varies and in presence of overlapping AP coverages.

2.7.1 Impact of vehicular communication technology adoption

As a first step in the evaluation of vehicular content downloading, we look at the impact that the diffusion of I2V and V2V communication technologies has on the system performance. To that end, we consider different values of p as well as different extensions of the roadside AP deployment. For clarity, we focus on the urban scenario depicted in Fig. 2.4 and we consider the AP deployment obtained by solving the max-flow problem on the full and the sampled DNTG. Also, we study non-overlapping AP coverages and constrain V2V relaying to 2 hops from APs; these assumptions will be relaxed later on.

Fig. 2.5 portrays the evolution of the key performance metrics when the technology penetration rate, p, varies between 5% and 80%. The curves refer instead to different



(a) Throughput (b) Delay

(c) Fairness (d) V2V download fraction

Figure 2.5. Max-flow strategy: Average per-downloader throughput (a), delay (b), fairness (c), and V2V downloading fraction (d) vs. p, for different AP deployment extensions

extensions of the roadside infrastructure, ranging from $\hat{A} = 15$ to $\hat{A} = 60$ APs. Note that the latter value essentially corresponds to a complete coverage of the road topology by the APs. The results obtained using full and sampled DNTG are denoted by thick and thin lines, respectively.

Throughput. The average per-downloader throughput, in Fig. 2.5(a), is very satisfying in all conditions, scoring well above 10 Mb/s even in low-p, low-$\hat{A}$ scenarios, and more than 20 Mb/s in presence of a wide I2V and V2V technology adoption. When separating the effects of $\hat{A}$ and p, the availability of a more pervasive (although non overlapping) infrastructure coverage helps at both low and high-penetration rates, although its impact is lower than one could expect. Indeed, a pervasive 60-AP deployment only results in a constant 3-Mb/s gain over a simple 15-AP deployment. Higher improvements can be instead obtained from the spread of in-car communication interfaces, with an average throughput increase of 8 Mb/s as p grows from 5% to 80%. We remark that

the steepest throughput growth lies between 5% and 20% penetration rate.

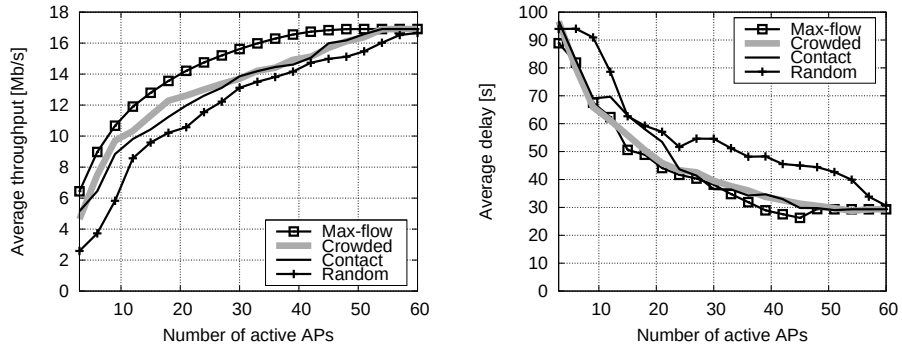
Delay. Delays, in Fig. 2.5(b), are in the order of tens of seconds, a good result when considering the delay-tolerant nature of most V2V transfers. We can observe different behaviors for low and high values of p. For $p < 20\%$, an increase in availability of relays leads to more frequent V2V transfers, hence higher delays. When $p > 30\%$, the already pervasive presence of relays makes the impact of even higher penetration rates negligible. Also, a denser AP deployment helps reducing the delay, although such a gain is significant only when $\hat{A}$ is low.

Fairness. To get an insight on how the system throughput is actually shared among the downloaders, Fig. 2.5(c) shows the Jain's fairness index. The increase in penetration rate has a major impact since it implies a growing number of V2V communication opportunities. Indeed, for low values of p, downloaders travelling over secondary roads have fewer chances to benefit from traffic relay than downloaders travelling on main (typically, more crowded) roads. It follows that some unfairness arises for low penetration rates, while the system becomes fair for medium-high values of p. Also, the larger the $\hat{A}$, the higher the level of fairness, as both main and secondary roads can be covered.

Transfer paradigm. The above observations on the fundamental role of V2V traffic relaying is confirmed by the results in Fig. 2.5(d), depicting the fraction of content downloaded through relay vehicles. Indeed, most of the content is received by downloaders from relays (through either connected forwarding or carry-and-forward). Clearly, the importance of V2V communication tends to grow with the penetration rate p, since the availability of additional relays allows a more intensive utilization of the wireless resources. More surprisingly, the presence of additional APs only marginally reduces the utilization of V2V communication, and, at high values of p, more than 80% of the data is downloaded through relays even when the whole road surface is covered by APs. We will further comment on this phenomenon later in this section.

Problem solution. Fig. 2.5 highlights the effectiveness of the sampling-based technique introduced in Sec. 2.5.3, when compared against the optimization solution on the full DNTG. The performance results obtained with the latter are shown in all the plots as thick grey curves, while the thinner lines represent the outcome of the sampling-based solution. The throughput and delay loss induced by the sampling are negligible, and the fraction of V2V downloading is identical in the two cases. The only noticeable difference can be observed in terms of fairness, since, by sampling the vertices representing the candidate AP locations with higher weight, APs on secondary roads are seldom activated in the max-flow solution, thus reducing the level of fairness.

We remark that sampling the DNTG allows to solve significantly more complex instances of the max-flow problem. As an example, in the plots of Fig. 2.5, memory requirements become too demanding for the solution of the complete problem when more than 20% of the vehicles are part of the network.



(a) Throughput (b) Delay

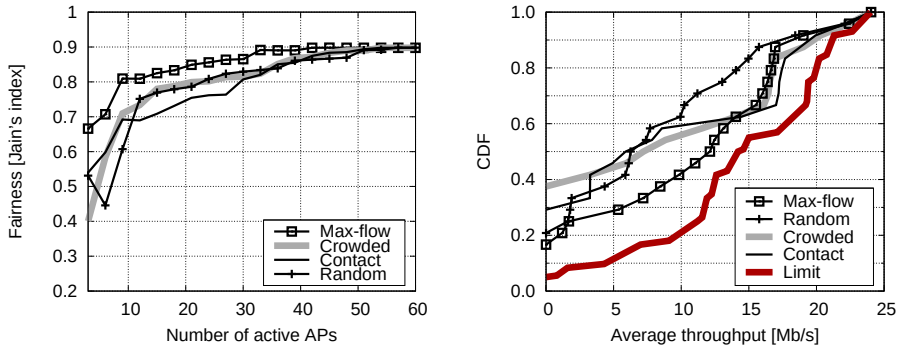
Figure 2.6. Low-penetration regime: Average per-downloader throughput (a) and delay (b), vs. $\hat{A}$, for different AP deployment strategies

Summary. The performance metrics are consistent in revealing the critical importance of the penetration rate p and the lower impact of the roadside infrastructure extension. Accordingly, we can separate two regimes. The first, when $p < 20\%$, i.e., at early stages of the technology adoption, characterized by lower throughput and higher delay, a stronger dependency on direct I2V communication and lower downloading fairness. The second, for $p > 30\%$, i.e., in presence of a quite mature technology, featuring instead higher throughput and lower delay, massive use of V2V communication and high fairness. As the impact of the system settings is different within these two regimes, in the following we will study them separately. According to the results above, we will employ the max-flow problem solution on the complete and on the sampled graph in the low- and high-penetration regime, respectively.

2.7.2 Low-penetration regime

As case study of the low-penetration regime, we consider $p = 10\%$. The default settings include the urban scenario, non-overlapping AP coverages, a 1% fraction of downloaders and a 2-hop limit in V2V relaying.

AP deployment. Fig. 2.6 shows the average per-downloader throughput and delay for the different deployment strategies, as the number of active APs $\hat{A}$ varies. Overall, the performance at early deployment stages is satisfactory. The plots confirm that increasing $\hat{A}$ positively affects the downloading performance. However, it is also clear that the extension of the infrastructure deployment is more critical when the number of active APs is low. Indeed, activating 20 APs yields a 8-Mb/s throughput and 35-second delay gain over a 5-AP deployment, while the activation of additional 40 APs only leads to mere 3-Mb/s throughput and 15-second delay improvement.



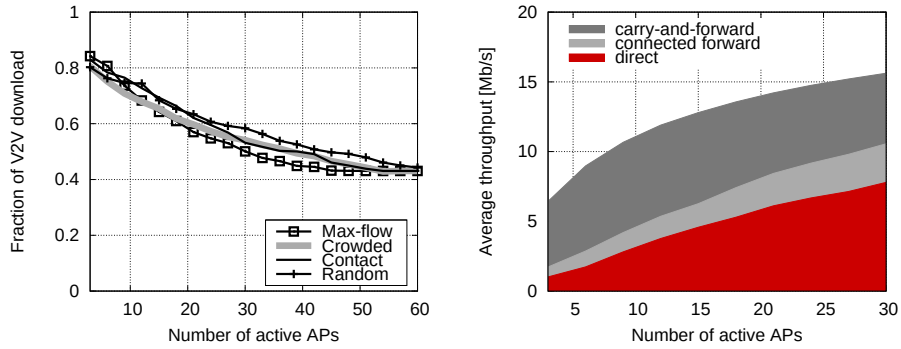
(a) Fairness (b) Throughput CDF

Figure 2.7. Low-penetration regime: Fairness (a) and CDF of the per-downloader throughput when $\hat{A} = 15$ (b), for different AP deployment strategies

The figure also highlights the impact of different AP deployment techniques in the low-penetration regime. As one can expect, the AP placement dictated by the Max-flow strategy guarantees the best performance in terms of both throughput and delay, while a Random deployment of APs yields the worst result. AP placements based on the Crowded and Contact approaches fall in between. If the performance ranking of the deployment strategies is constant throughout all values of $\hat{A}$, the same is not true for the relative gain. Indeed, when a few APs are activated, a well-planned deployment can result in a 200% throughput gain over a random placement. As the number of deployed APs grows, such an advantage is progressively reduced: in particular, when APs cover more than 50% of the road topology (i.e., $\hat{A} > 30$), using non-optimal approaches to AP deployment makes the performance quickly close in towards those achieved with a random placement. Finally, as the active APs tend to cover the whole region, optimal and non-optimal strategies yield similar performance.

Fairness. As shown in Fig. 2.7(a), the system favors downloaders travelling on the main roads when $\hat{A}$ is low, while user experience tends to be leveled as more and more APs are activated. The Max-flow deployment results in a slightly fairer system, while no significant difference can be appreciated among non-optimal placement strategies.

The reason for the unfairness for small AP deployments is investigated in Fig. 2.7(b), for $\hat{A} = 15$. The plot reports the CDF of the per-downloader throughput, and shows a large heterogeneity in the amount of content obtained by different users. On the one hand, a significant percentage of downloaders, between 20% and 40% depending on the deployment strategy, experiences zero throughput. On the other, the luckiest 10% of downloaders enjoys a throughput ranging between 16 and 20 Mb/s.



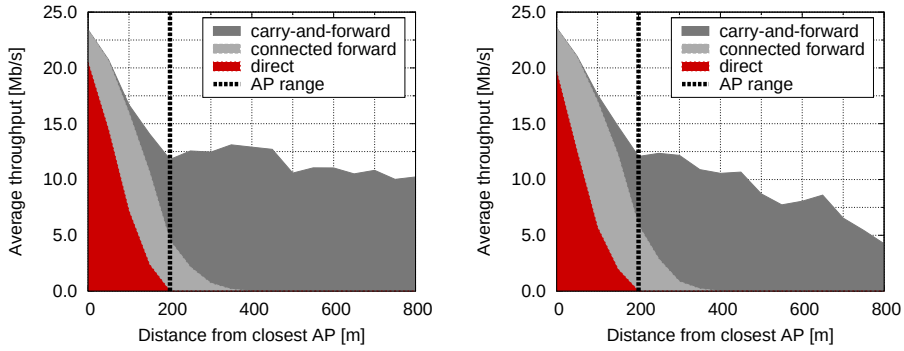
(a) V2V download fraction (b) Paradigm split-up

Figure 2.8. Low-penetration regime: Fraction of data downloaded through V2V (a) and transfer paradigm split-up of the per-downloader throughput (b). In the latter plot APs are deployed according to the Max-flow strategy

Although fairness in content downloading is not an objective of the max-flow problem[4], we point out that such unfairness is only marginally due to our formulation. Rather, it can be attributed to the very different conditions incurred in by downloaders in a realistic mobility trace, such as various traffic intensity on the roads they travel on, or different time intervals spent within coverage of the APs in their trip. This is confirmed by the limit curve in Fig. 2.7(b), which presents the CDF of the maximum achievable throughput, i.e., the throughput that each of the downloaders in the trace would experience if it were the only downloader in the network, with all resources and relays at its disposal. Not even in such ideal conditions one can guarantee fairness among all users, given their different trips. More pervasive AP coverages (Fig. 2.7(a)) or additional relaying opportunities (Fig. 2.5(c)) can help to reduce disparities, by providing transfer paths to downloaders travelling on secondary routes.

Transfer paradigm. The fraction of content downloaded through vehicular relays is shown in Fig. 2.8(a). Across almost all AP deployment strategies, the V2V downloading fraction is around 0.8 when a low number of APs is activated, and then decreases as $\hat{A}$ grows, i.e., as a direct access to the infrastructure becomes more pervasive. This is a rather intuitive behavior that, however, yields an interesting result when coupled with the average per-downloader throughput, as in Fig. 2.8(b). There, we can observe that the fraction of content downloaded through V2V relaying is somewhat constant, contributing approximately 5 Mb/s to the overall throughput, regardless of the number of deployed APs. Also, the dominant relay paradigm is carry-and-forward.

[4]We remark that, in an attempt to provide the downloaders with fairer performance, we have considered a max-min formulation instead of the max-flow one. However, due to the diversity in the downloader conditions highlighted next, no minimum positive throughput could be guaranteed.



(a) Max-flow (b) Crowded

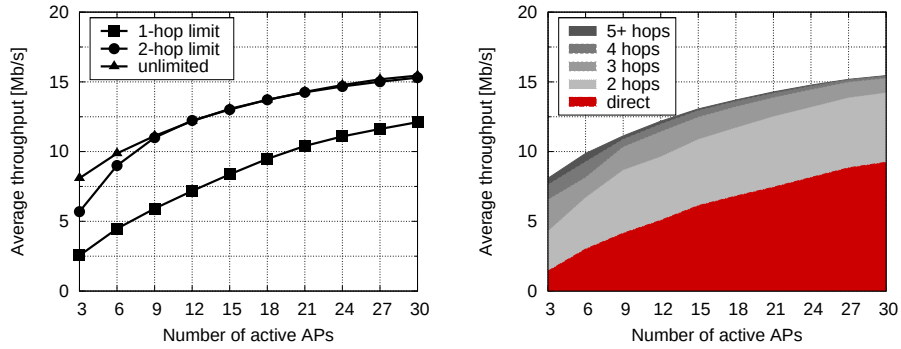
Figure 2.9. Low-penetration regime: Average per-downloader throughput as a function of the distance between the downloader and the closest AP

A quite surprising result is that, even when APs fully cover the road topology (for $\hat{A} = 60$), V2V relaying is still widely employed in the downloading process. The reason behind such a phenomenon is unveiled in Fig. 2.9, portraying the average per-downloader throughput as a function of the downloader distance from the closest AP, for the Max-flow and Crowded (i.e., the most performing) strategies. The plot highlights the portions of traffic transferred through the different paradigms. As one would expect, direct I2V transfers can only occur within the AP transmission range, and a 2-hop connected forwarding reaches at most twice such a distance. Carry-and-forward is not distance-bounded, hence it can reach downloaders that are very far from APs.

However, a key observation is that V2V relaying frequently occurs within range of APs. In fact, at a distance of 100 m, i.e., half of the maximum transmission range of the AP, communication largely takes place through relays. The reason is that our model realistically accounts for the network-layer rate decrease with distance, hence making the use of high-rate multi-hop paths preferable to low-rate direct transfers. This explains why, even in presence of a pervasive AP coverage, relaying is employed to improve the wireless resource utilization and, thus, the overall throughput.

Unlimited relaying. All previous results assumed a 2-hop limit in data transfers, basically constraining V2V relaying to one hop at most. We now relax this assumption and compare three different scenarios, where (i) only direct I2V communication is allowed, (ii) the 2-hop limit is enforced, and (iii) unlimited relaying is allowed.

Fig. 2.10(a) depicts the average throughput achieved in the three cases, when the APs are deployed according to the Max-flow strategy. It is clear that, in absence of relaying through vehicles, downloaders can only leverage direct contacts with the APs, which leads to a significantly lower throughput. Allowing a single relay between APs and downloaders yields a throughput gain between 150% and 20%, depending on the



(a) Throughput (b) Paradigm split-up

Figure 2.10. Low-penetration regime: Average per-downloader throughput (a) and transfer paradigm split-up (b), when APs are deployed according to the Max-flow strategy. In the latter plot, the number of relay hops in unconstrained

coverage provided by the infrastructure. Even more interestingly, considering transfers over 3-hop long or more yields almost no advantage over the case where a 2-hop limit is enforced.

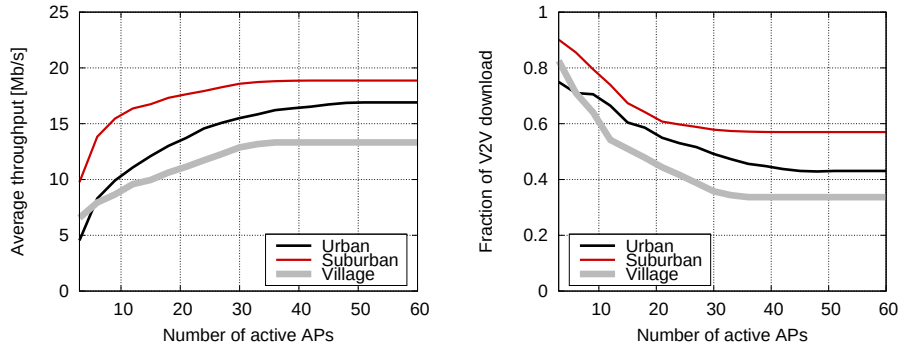
In order to explain the latter effect, we fragment the downloading throughput measured in the former scenario, according to the number of hops traveled by packets to reach their destination. Fig. 2.10(b) shows how, even when unlimited hops are allowed, a large majority of the relayed data traffic arrives at destination in just two hops. There is a small probability of going through 3 hops, while 4 hops or more are almost never employed. Indeed, when comparing Fig. 2.10(b) to Fig. 2.8(b), it is clear that the availability of additional hops, which grants more flexibility to the max-flow problem solution, only leads to minor adjustments that have a negligible impact on the overall downloading performance.

Road environment. The average per-downloader throughput recorded in the three road topologies presented in Sec. 2.6 is portrayed in Fig. 2.11(a), when the APs are deployed as dictated by the Max-flow strategy.

The overall performance trend in the new environments is the same as already observed in the urban scenario, thus our considerations on the impact of the AP deployment also hold for the village and suburban environments.

However, we can observe that the relative result in suburban and village environments differs from that measured in the urban case. On the one hand, the throughput in the village scenario is lower than in the urban one, with a significantly reduced utilization of V2V relaying. On the other, the suburban scenario yields higher V2V download fraction and per-downloader throughput.

The reason for these different behaviors is found in the diverse nature of vehicular



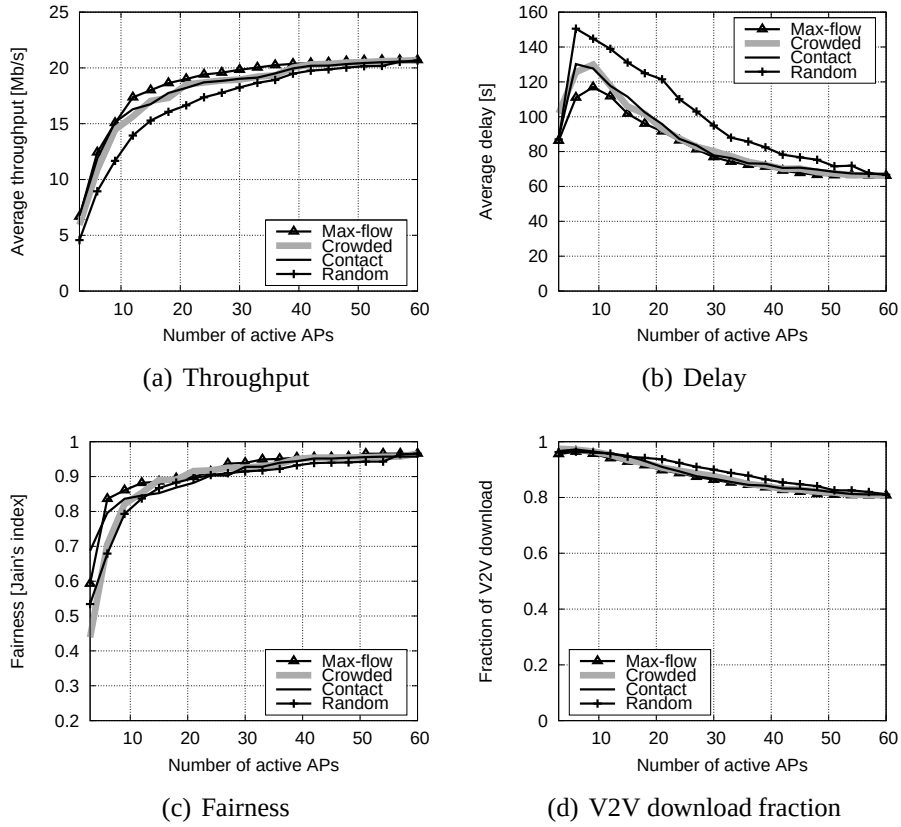
(a) Throughput (b) V2V download fraction

Figure 2.11. Low-penetration regime: Average per-downloader throughput (a) and fraction of data downloaded through V2V (b) vs. $\hat{A}$, in different road environments. APs are deployed according to the Max-flow strategy

traffic in the tree regions. By looking at Fig. 2.4(b), it is clear that fewer vehicles circulate in the village environment than in the urban one: thus, for a given p, fewer vehicles participate in the content downloading as relays. Moreover, the traffic is distributed over the road topology quite evenly, which makes it difficult to find an AP deployment that well covers most of the vehicular traffic. As a result, downloaders in the village scenario are penalized in terms of throughput.

In the suburban scenario, the car traffic volume is close to that observed in the urban environment, which means that the number of available relays in the two cases is similar. However, the suburban region is characterized by a few high-traffic thoroughfares and many low-traffic secondary roads. As the vehicular traffic is so concentrated, it is easier to deploy a few APs in the right locations; also, downloaders have higher chances of meeting many relays on their way. Thus, drivers in the suburban environment typically enjoy a higher throughput.

Summary. In the low-penetration regime, the early infrastructure deployment stages are critical. When just a few APs are activated, the policy chosen for their placement has a major impact on the user experience, as optimal deployments lead to a throughput twice or three times higher than that observed with careless placements. Moreover, the activation of a few more (or less) APs dramatically affects the throughput, delay and fairness of the system. Since the downloading performance during early adoption phases will play an important role in attracting new users, the AP deployment should be carefully studied when introducing the technology. Conversely, the placement of too many APs may have a small impact on the downloading performance, while significantly increasing the deployment and maintenance costs.



(a) Throughput (b) Delay

(c) Fairness (d) V2V download fraction

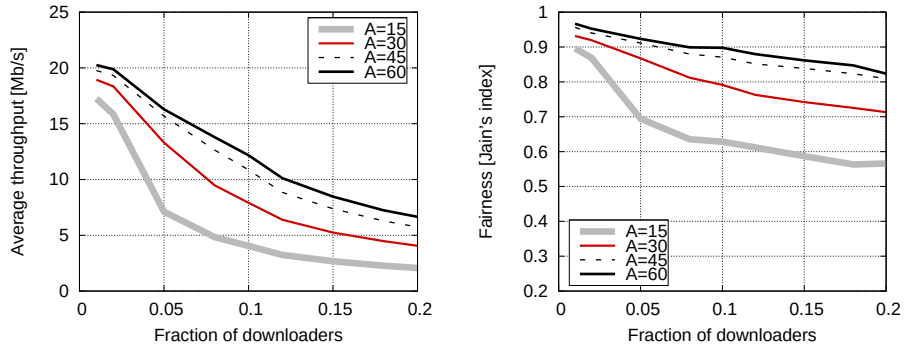
Figure 2.12. High-penetration regime: Throughput (a), delay (b), fairness (c), and V2V downloading fraction (d) vs. $\hat{A}$, for different AP deployment strategies

One should not expect the system to be fair, or to have similar performance in different road environments, since the diversity in the routes traveled on by drivers lead to intrinsic differences in their download experience.

As a final remark, our results suggest that the complexity of designing multi-hop relaying protocols can be safely avoided, by limiting the process to one relay, without incurring in performance penalties. This confirms recent findings on bus networks [15, 21], which thus apply also to a more general vehicular downloading context.

2.7.3 High-penetration regime

In the high-penetration regime, we consider $p = 50\%$ and the max-flow problem is solved on the sampled DNTG. Once more, the default settings include the urban scenario, non-overlapping AP coverages, a 1% fraction of downloaders and a 2-hop limit in V2V relaying.



(a) Throughput (b) Fairness

Figure 2.13. High-penetration regime: Average per-downloader throughput (a) and fairness (b) vs. d, under the Max-flow strategy and for different AP deployment extensions

AP deployment. The overall performance is outlined in Fig. 2.12, for different extensions and strategies of the roadside infrastructure deployment. When comparing the results to those obtained in the low-penetration regime, we observe a significant improvement in the absolute value of the throughput, in Fig. 2.12(a), that now reaches more than 20 Mb/s – a clear effect of the increased availability of relays. The throughput growth is much faster as additional APs are deployed, with nearly optimal performance attained with as few as 15 active APs. This is due to the fact that relays can now easily compensate for undersized infrastructure, as also demonstrated by the extremely frequent utilization of V2V communication, in Fig. 2.12(d), employed in 80% to 98% of the transfers.

As far as delay is concerned, Fig. 2.12(b) exhibits a peculiar behavior. Given the high number of users, several downloaders happen to travel on secondary roads. For a very low number of APs, such roads are scarcely covered, hence a number of downloaders experience zero throughput. Their delay is not accounted for, and the dominant contribution is limited to the few lucky fast downloaders. As the deployment becomes more widespread and $\hat{A}$ increases, more downloaders experience non-zero coverage time, including those on secondary roads where the chances to carry on the download are few and far between. For even denser deployments, such delays are mitigated by the availability of more APs.

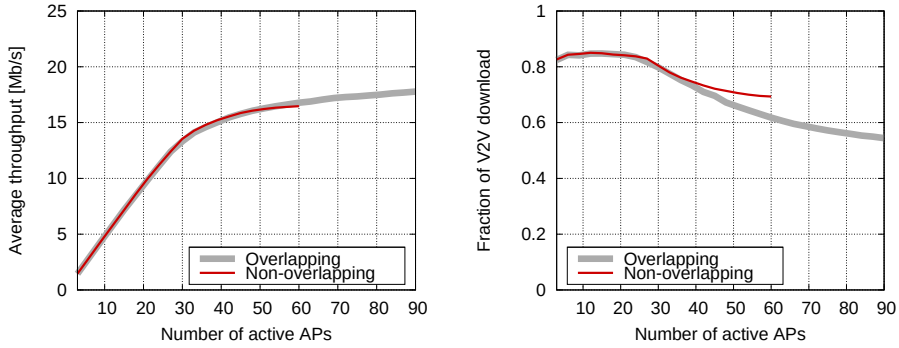
Finally, the massive presence of relays helps to reduce the unfairness, in Fig. 2.12(c), as downloaders have high chances to meet relays, regardless of the route they take.

Concurrent downloaders. In presence of a wide diffusion of I2V and V2V communications, the downloading activity by users participating in the system is likely to grow. Thus, in the high-penetration regime, it is important to evaluate the impact of the

amount of concurrent downloaders, i.e., users requesting some content during the same time interval.

In Fig. 2.13, we observe the impact of the concurrent downloader fraction d on the system performance. We consider d ranging between 0.01 and 0.2, the latter representing a highly-loaded system, in which one out of five users is downloading some content at any time instant.

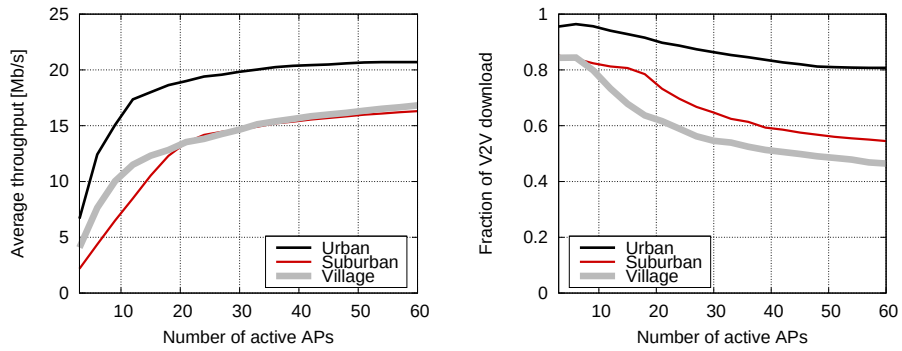
As one could expect, when the system load grows, increasing the number of APs comes in handy, and can noticeably improve throughput (Fig. 2.13(a)) and system fairness (Fig. 2.13(b)). Also, increasing the demand (especially, for $d > 0.1$) reduces the per-user throughput, due to the augmented contention for the limited wireless resources. Less intuitively, fairness degrades as d grows: when the number of simultaneous downloaders increases, vehicles travelling on secondary roads experience less channel contention, hence higher throughput than vehicles travelling on main (more crowded) roads.



(a) Throughput (b) V2V download fraction

Figure 2.14. High-penetration regime: Throughput (a) and fraction of data downloaded through V2V (b) vs. $\hat{A}$, when $d = 0.05$. The case of overlapping and non-overlapping AP coverages are compared, under the Max-flow strategy

Overlapping AP coverages. Given the beneficial effect of additional APs when p is high, we study the impact of a further infrastructure extension by allowing AP coverages to overlap. Fig. 2.14 compares the throughput and V2V download fraction obtained when non-overlapping and overlapping AP coverages are allowed. The results have been obtained for a relatively high downloading demand, namely, $d = 0.05$. When overlapping among AP coverages is not allowed, only 60 candidate locations can be considered. Conversely, such a number grows to 90 when the coverage of any two candidate APs can overlap. Observe that, for a fixed number of APs, the possibility to have overlapping coverages leads to a marginal improvement in the per-downloader throughput. As shown by Fig. 2.14(b), the reason for this behavior is once more that the V2V traffic relaying tends to compensate for the lack of flexibility of the non-overlapping



(a) Throughput (b) V2V download fraction

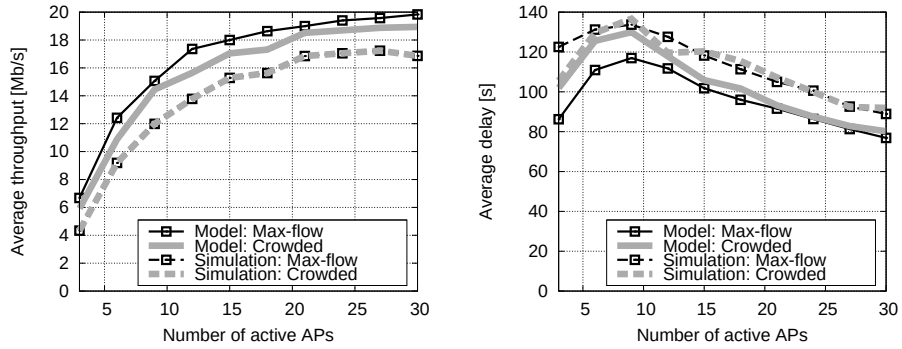
Figure 2.15. High-penetration regime: Average per-downloader throughput (a), and fraction of data downloaded through V2V (b) vs. $\hat{A}$, in different road environments. APs are deployed according to the Max-flow strategy

deployment.

Road environment. Fig. 2.15 shows the throughput and V2V download fraction in the urban, suburban and village scenarios. As already observed in the low-penetration regime, also in presence of high p the road topology has a major impact on the downloading process. However, the relative performance of the three scenarios are different with respect to those in Fig. 2.11. The highest throughput is now achieved in the urban scenario, while drivers in the suburban and village environments experience similarly worse performance.

The reason lies in the increased contention for resources, induced by the higher participation of vehicles in the network. In the urban scenario, many vehicles travel over different roads, which basically allows a spatial reuse of the wireless medium. The village scenario is similar to the urban one, in that vehicular traffic is quite evenly spread over the road topology; however, the lower number of vehicles reduces the availability of relays, as also evident from Fig. 2.15(b). In the suburban scenario, instead, a high vehicular density is concentrated on a few roads: the consequent channel congestion yields reduced per-downloader throughput.

Summary. The analysis in the high-penetration regime significantly differs from the early technology adoption phase. When the technology is mature and spread enough, the infrastructure deployment will play a minor role, and a few, randomly deployed APs will suffice to achieve near-optimal downloading performance. Indeed, V2V communication will be able to sustain the system, no matter the underlying AP placement. Pervasive non-overlapping APs will be needed only in case the technology attains a level of success such that the number of concurrent downloaders grows well above the percentages today's recorded in wired networks. In this case, channel contention will



(a) Throughput (b) Delay

Figure 2.16. High-penetration regime: Average per-downloader throughput (a) and delay (b) obtained through model and simulation, as the number of APs and the deployment strategy vary

become the primary constraint to the downloading performance, with I2V and V2V transfers contending for air time across the whole road topology. As such, redundant coverages will not yield a significant throughput gain and downloaders travelling on more crowded roads will experience worse performance.

2.8 Impact of MAC and physical layer modeling

The max-flow problem we formulate relies on a simplified model of channel access and RF signal propagation. Since our goal is to derive an upper bound to the performance achievable in a real-world deployment, these assumptions are not especially limiting. However, one may wonder about the impact that more realistic MAC and physical layer representations have on the system, i.e., how much their idealization contributes to shift the upper bound away from the actual performance.

Including complex models of signal propagation and layer-2 protocols in the optimization problem is unfeasible, thus we rely on simulation to evaluate these aspects. More precisely, we employ ns-3, due to its remarkable accuracy in modeling both the physical and MAC layers, including the SINR and the bit error rate computation. At the physical layer, we adopt a log-distance propagation loss model with exponent 3.0; the transmit output power is set to 16 dBm. At the MAC layer, we use IEEE 802.11a with the AARF rate adaptation algorithm [29]. For each scenario, we feed the optimal scheduling to the simulator, and observe the performance at each downloader.

Fig. 2.16 depicts the average throughput and delay in the case of high-penetration regime. We report the results of the max-flow problem on the sampled DNTG and those obtained under the Crowded strategy. The optimization problem results are compared to

ns-3 simulations. It is evident that the difference in both throughput and delay is limited in all cases, which demonstrates the reduced impact that MAC and physical layers have on the overall performance. Indeed, it is the optimality of the scheduling that plays the most important role in determining the downloading experience of the users, rather than the local channel access coordination of individual transmissions.

Similar results have been obtained for the less critical case of the low-penetration regime.

Moreover, we stress that plots are limited to 30 APs along the x-axis, since larger simulations would have required an exceedingly long computational time. This further underscores the usefulness of our formulation, which makes much more extensive evaluations of the downloading system feasible.

2.9 Conclusions

We proposed a novel framework based on time-expanded graphs for the study of content downloading in vehicular networks. Our approach allows to capture the space and time network dynamics, and to formulate a max-flow problem whose solution provides an upper bound to the system performance. Through a graph-sampling technique, we solved the problem in presence of realistic, large-scale traces, and we analysed the impact of several key factors on the performance limits. Simulation results showed that the physical- and MAC-layer assumptions on which the framework relies have a minor impact, leading to a tight upper bound.

The major findings in our analysis are as follows.

(i) Two separate regimes, characterized by different performance and impact of the system settings, emerge at different technology penetration rates. In a typical urban scenario, the watershed arises when 20-30% of the vehicles participate in the network.

(ii) The strategy and the extension of the AP deployment play a major role in the low-penetration regime, with well-planned deployments leading to a throughput twice or three times higher than that observed under a careless placement. In the high-penetration regime, instead, even a random AP deployment works well and the pervasiveness of the APs becomes important only in presence of high downloading demand.

(iii) The contribution to performance of V2V traffic relaying is of fundamental importance. It can compensate for reduced coverage as well as for a non-optimal AP placement, with such an effect becoming more and more evident as the technology penetration rate increases. Interestingly, the contribution of V2V communications remains relevant even under a pervasive AP deployment and in both penetration regimes, as optimal scheduling tends to favor high-rate V2V transfers over low-rate I2V communications.

(iv) Knowledge of user mobility is critical to the system performance, since most of the V2V traffic relaying takes place through the carry-and-forward paradigm. However,

the complexity of multi-hop protocols can be limited to one relay, as the contribution of transfers over a higher number of hops is negligible. An interesting direction for future research is therefore the design of protocols that let the roadside infrastructure acquire accurate estimates of the vehicles encounter opportunities, and the definition of a scheduling algorithm that effectively leverages such information. We remark that, by using edges with probabilistic instead of deterministic weights, our graph-based model could be extended to account for the uncertainty in the mobility estimates, and to evaluate its impact on the system performance.
(v) The structure of the road topology and the route followed by vehicles determine the downloading performance experienced by the users. Thus, one should adapt the system configuration to the characteristics of the road environment. In any case, some unfairness should be expected unless there is a pervasive presence of APs and relays, and the number of downloaders is not overwhelming.

Chapter 3

Dealing with uncertainty

3.1 Introduction

In the previous chapter, we have has established that, in order to efficiently support content downloading, (i) RSU deployment should target the areas expected to be most crowded by vehicles and (ii) I2V content transfer should be complemented by V2V data relaying.

A part of the picture is still missing, though. Given the ability to delivery information to passing by vehicles through a carefully planned-out RSU deployment, what exactly should be delivered to them? A spotty coverage could meet expectations only on condition that the short time under coverage is fruitful: RSUs should prefetch the content so as to have it promptly available for passing-by vehicles requesting it. Matching between storage at RSUs and demands by vehicles is, however, easier said than done. One possibility is that RSUs have access to the content demand and to predictions of mobility patterns, and exploit them to take prefetching decisions, as in [16]. Additionally, to make V2V transfers more effective, RSUs can leverage a similar approach for I2V communication toward relay vehicles deemed to meet downloaders later on.

In order to relieve the cellular network from the content delivery task, our work is the first to jointly study the problems of content prefetching at RSUs, scheduling of I2V transmissions and management of V2V relay transfers, in presence of inaccurate mobility prediction.

To do so, we model the uncertainty affecting the mobility prediction through a *fog-of-war* probabilistic representation of the inter-node contacts. Such a model can provide an abstraction of any prediction technique and allows us to draw conclusions of general validity (Sec. 3.3). The output of the fog-of-war model is used to build a time-expanded graph with probabilistic weights, representing the evolution of the inter-node contacts (Sec. 3.4.1). We exploit the graph to formulate an optimization problem, to be solved at each RSU, that jointly addresses content prefetching and scheduling (Sec. 3.4.2). The

data scheduled by RSUs toward relays are then delivered to downloaders, according to different schemes, namely, a greedy strategy exploiting opportunistic encounters and a RSU-driven scheduling of relay-to-downloader transmissions (Sec. 3.5). In our performance evaluation, we compare the offloading efficiency of the system outlined above against benchmark solutions. Furthermore, we account for the impact of the social behavior of vehicular users on relay-based content transfers, and assess its benefits on the capability of ITS to relieve the cellular network (Sec. 3.6).

3.2 System model

We consider a DSRC-based vehicular network composed of mobile users and fixed roadside units (RSUs), deployed over a road topology that is also covered by a cellular infrastructure. As depicted in Fig. 3.1, RSUs provide a spotty but high-throughput, inexpensive connectivity to vehicles, whereas the cellular network guarantees seamless coverage, which however comes at some connection cost.

Users of the vehicular network may become *downloaders*, i.e., they may wish to retrieve different types of data from the fixed network (e.g., the Internet). Assuming that vehicles have both a DSRC and a cellular radio interface, multiple transfer paradigms for content delivery are possible. More precisely, downloaders can exploit the ITS network to perform *direct* transfers from the RSUs, or to be assisted by other vehicles acting as *relays*. In the latter case, we consider connected forwarding, i.e., traffic relaying through a connected multi-hop path, as well as carry-and-forward, i.e., traffic relaying through vehicles that store and carry the data before delivering them to the target downloader. Alternatively, downloaders can resort to *cellular* transfers, in order to retrieve the desired content from the fixed network.

We model the downloaders' demand by considering *what* they request and *how* they get it, as follows. As far as the *what* is concerned, we account for the fact that users may belong to social groups, characterized by, e.g., partially matching mobility patterns or long-lived contacts. Vehicular users of a same group request contents with the same probability distribution, so as to reflect the correlation between a user's social group and the information it requests [30]. Regarding the *how*, downloaders try at first to obtain the data through inexpensive opportunistic exchanges with RSUs and relay vehicles. If the desired content cannot be fully retrieved within a time deadline T, the downloaders will pay to fetch the remaining portion via a cellular transfer. Note that this model provides an incentive for users to offload the cellular network through ITS.

Next, we detail the operations that the network and the users undertake during the content downloading process.

A user wishing to retrieve a content generates a request to an Internet-based query management system, via either an RSU or the cellular network [16]. Such a management system forwards the pending request to the RSUs in the area where the downloader

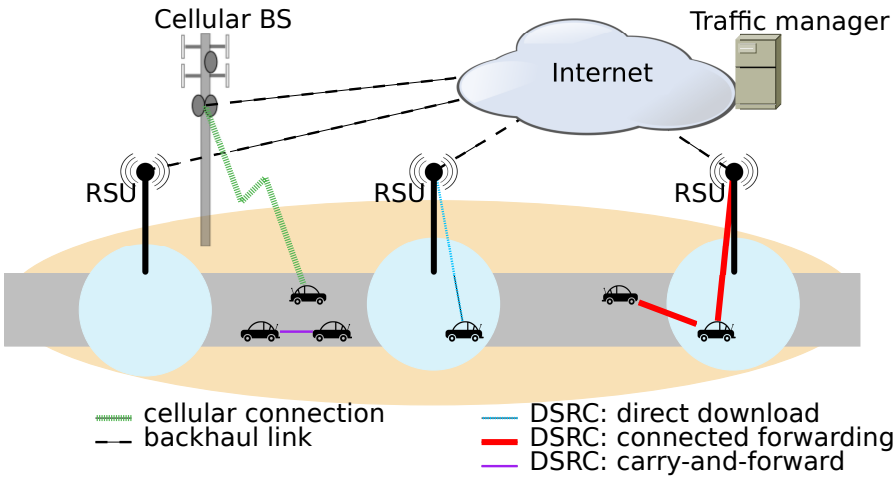


Figure 3.1. Network system

is traveling. RSUs are then in charge of (i) fetching portions of the content from some server storing it, and (ii) delivering the data to the target downloader directly, or to a relay vehicle deemed to meet the downloader later on.

It is clear that, in order to efficiently use the network resources over the backbone and the airtime on the wireless medium, RSUs must take content prefetching and scheduling decisions by foreseeing future direct or relay transfer opportunities that involve downloader vehicles [31, 32]. To that end, we assume that a forecast of the future I2V and V2V contacts is periodically issued by a traffic manager to the RSUs, as in emerging real-time traffic monitoring systems [33]. Such information also includes the identity and pending queries of downloaders that are in the network at the time of the issued forecast; we stress that the traffic manager is instead unaware of future content requests.

Based on such contact information and taking into account the rate B at which data can be retrieved from the server, the RSUs make locally optimal decisions on which data to prefetch and toward which vehicles (either relays or downloaders) they should be transmitted. If RSUs delegate portions of content to relays, and these are in range of, or subsequently meet, a downloader interested in such a content, V2V transfers occur. Multi-hop data transmissions, be they of the connected forwarding or carry-and-forward type, are limited to two hops from the RSU, since this already allows for nearly optimal performance [23]. We also remark that all vehicles are assumed to be available to relay traffic whenever they are not receiving data from an RSU. Finally, given the storage capabilities of today's communication nodes, the memory capacity at RSUs and vehicles is not considered to be an issue.

3.3 The traffic manager prediction

We assume that the traffic manager predicts the node mobility with time granularity δ; in the following, we refer to the interval of duration δ as time step. The prediction is updated every H steps, upon the reception of new information on vehicle positions. We also consider that H is the time horizon over which the prediction is made.

Based on the predicted positions of vehicles, the traffic manager considers that two nodes (either mobile users or RSUs) are neighbors if their distance is below or equal to their maximum radio range. Then, by defining a wireless link shared by a pair of neighboring nodes as a *contact*, it forecasts the contacts existing at each of the next H time steps. A contact may extend over multiple steps and its data rate can be related to the node distance and propagation conditions (see, e.g., the description in Sec. 3.6.1).

To model the limited accuracy in the prediction of the contacts and their characteristics, as compiled by the traffic manager, we adopt a prediction technique-independent approach. Rather than considering one specific prediction methodology (e.g., among those cited in Sec. 3.7), we propose a *fog-of-war model*, which provides an accurate abstraction of virtually any prediction technique and accounts for different precision levels of the forecast [34].

More specifically, let $\mathcal{P}(u,H)$ be a contact prediction generated by the traffic manager at step u for the next H steps. Given that the prediction accuracy may be affected by several sources of error, we assume actual V2V and I2V contacts occurring between the present time, u, and the prediction horizon, $u+H-1$, to be affected by a Gaussian-distributed noise with zero mean and variance σ^2. More formally, for each contact between a generic node pair starting at step $k \in [u,u+H)$, we extract a realization ν of the noise. If $|\nu| \leq 1$, we associate a probability $1-|\nu|$ to the contact, which expresses the likelihood with which the traffic manager expects the contact to take place. Otherwise, the contact is evicted and a new, *spurious* one is created and associated with a probability equal to $\min\{|\nu|-1,1\}$. The nodes sharing the spurious contact are chosen randomly among the network nodes and inherit the duration and data link rate of the true contact that it has replaced. This simple model allows us to capture the possibility that prediction techniques underestimate actual contact opportunities, when $|\nu| \leq 1$, and wrongly forecast future contacts, when $|\nu| > 1$.

The variance σ^2 models the accuracy of the prediction, since the larger the zero-mean noise variance, the less precise the estimation of the connectivity. We express the variance as $\sigma^2 = \sigma_0^2(k-u)$ for V2V contacts and $\sigma^2 = \frac{\sigma_0^2}{2}(k-u)$ for I2V contacts. Indeed, due to the mobility of both link end-points, we expect V2V contacts to be affected by a variance that is twice that of I2V contacts. Also, we let σ^2 grow linearly with $k-u$, which accounts for the fact that predicting contacts farther in time becomes increasingly harder. As a result, spurious contacts, appearing with the same frequency with which actual contacts are evicted, are more frequent if the prediction accuracy is

low (i.e., high σ_0^2) and the estimation is pushed far ahead in time (i.e., large $k-u$).

For each contact predicted by the traffic manager (be it correct or not), we also account for possible errors in the estimation of the duration and the time evolution of the link data rate. Specifically, we add to the current value of the contact duration a random number of steps uniformly distributed between $[-\tau,\tau]$, and we evict the contact if the obtained value is not positive. Likewise, we extract once for the whole contact duration a random value uniformly distributed in $[-\rho,\rho]$ and we add it to the link data rate computed at each step. Then, we ensure that the resulting value is neither negative nor greater than the maximum data rate. We remark that, by introducing some errors on the prediction of contact duration and data rate, our fog-of-war model also accounts for wrong estimates of the number of contacts by the traffic manager.

Finally, we point out that, since our fog-of-war model is defined by the values of σ_0^2, τ and ρ, by varying them, we can match the output of different prediction techniques. To verify that, we applied a Markovian prediction technique of the first and second order to the reference scenario that we use later in our performance evaluation (see Sec. 3.6.1). We found a very good agreement when σ_0^2=1.68, τ=23.92 and ρ=8.75, for the first-order model, and when σ_0^2=1.22, τ=18.32, and ρ=7.92, for the second-order model. Details on such an experiment can be found in [34].

3.4 Pre-fetching and scheduling at RSUs

Upon compiling the prediction $\mathcal{P}(u,H)$, the traffic manager forwards it to each RSU which, in turn, updates it with the contacts with passing vehicles it actually sees (whether they were predicted in advance or not). Such contacts are assigned a probability equal to 1, while wrongly predicted I2V contacts involving the RSU are assigned a zero probability. Thus, each RSU r_i has its own prediction $\mathcal{P}_i(u,H)$ and updates it as the time elapses. The prediction is used to generate a directed time-expanded graph with probabilistic weights (TEG-WP), on which the RSU formulates a linear programming (LP) problem that jointly optimizes prefetching and scheduling.

3.4.1 Building the TEG-WP

The prediction $\mathcal{P}_i(u,H)$ allows an RSU r_i to model the time evolution of the contacts between network nodes through a time-expanded graph. Since the prediction is based on discrete time steps of duration δ, the same granularity is used in the construction of the graph.

In the graph, each vehicle v_l appearing in the prediction $\mathcal{P}_i(u,H)$ at step $k \in [u,u+H)$ is associated to a vertex v_l^k, whereas each RSU r_i is mapped at each step k onto a vertex r_i^k. We denote by $\mathcal{V}^k$ and $\mathcal{R}^k$ the sets of vertices representing, respectively, the vehicles and the RSUs at step k. At every k, a directed edge connecting two vertices

represents the predicted contact between the corresponding pair of nodes. Such edges are referred to as intra-step and correspond either to I2V links, i.e., of the type (r_i^k,v_l^k), or to V2V links, i.e., of the type (v_l^k,v_m^k). We denote the set of I2V edges during step k by $\mathcal{E}_r^k$, and that of V2V edges by $\mathcal{E}_v^k$. Every intra-step edge in $\mathcal{E}_r^k$ and $\mathcal{E}_v^k$ is associated to a finite weight, representing the predicted data rate of the corresponding link at step k. As previously outlined, at the generic $k \in [u,u+H)$, each contact in $\mathcal{P}_i(u,H)$ is characterized by a probability of occurrence and an estimated data rate. We thus include these two aspects in the weight of an intra-step edge. As an example, consider a V2V contact between vehicles v_l and v_m at step k. We associate to the edge (v_l^k,v_m^k) a weight $w(v_l^k,v_m^k) = p(v_l^k,v_m^k) \cdot b(v_l^k,v_m^k)$, where $p(v_l^k,v_m^k)$ is the estimated contact probability between the two vehicles at k and $b(v_l^k,v_m^k)$ is the estimated maximum amount of data that can flow over the link during that time step. An identical discussion applies to I2V contacts.

Also, directed edges, of the type (v_l^k,v_l^{k+1}) or (r_i^k,r_i^{k+1}), are drawn between vertices representing the same node at two consecutive steps. While the edges in $\mathcal{E}_v^k$ and $\mathcal{E}_r^k$ represent anticipated transmission opportunities, these edges, referred to as intra-nodal, model the same node over time. They thus represent the possibility that vehicles physically carry data during their movement. Since we assume that the vehicle memory capabilities are unlimited, all intra-nodal edges are associated to an infinite weight. Note that accounting for the contact duration, instead of considering them as atomic, allows to model critical aspects of the real-world communication, like channel contention and the presence of hidden nodes.

Finally, the server(s) (from which RSUs retrieve the data) are modeled as a vertex named α. The graph is completed with B-weight edges (α,r_i^k), from α to any vertex $r_i^k \in \mathcal{R}^k$.

3.4.2 Making optimal decisions

At each step k, RSU r_i needs to take its prefetching and scheduling decisions. Specifically, each RSU determines: (i) which data, among those not already stored, have to be prefetched, in order to be transmitted to the vehicles (accounting for the limited rate at which data can be retrieved from the server); (ii) which data already available[1] at the RSU have to be delivered via I2V contacts actually seen at step k, i.e., to downloaders through direct transfers as well to candidate relays deemed to meet downloaders later on.

RSUs take decisions with the aim to maximize the user satisfaction. Given our scenario, where users first try to use the vehicular network and then fall back to cellular connectivity, we can assume that users satisfaction depends on: (i) the content delivery delay; (ii) the fraction of the content they retrieve through the vehicular network

[1] Data cached at RSUs are modelled by the flow on intra-nodal edges.

(i.e., within a time T since their request). Thus, each RSU formulates an optimization problem based on its TEG-WP as detailed next.

Let $\mathcal{C}$ be the content set, $t_{l,c}$ the step at which the generic downloader v_l sends a request for content c, and $\phi_{l,c}^k$ the fraction of the content that a user downloads at step k through the ITS network. Then, each RSU maximizes the following Aggregate Objective Function (AOF) over all content c and downloaders v_l:

$$\sum_{l}\sum_{c\in\mathcal{C}}\left[\sum_{k=t_{l,c}}^{t_{l,c}+T}\phi_{l,c}^k-\sum_{k=t_{l,c}}^{t_{l,c}+T}\phi_{l,c}^k\frac{k-t_{l,c}}{T}\right], \tag{3.1}$$

where the first sum within parenthesis is the total fraction of content downloaded by the user from the vehicular network and the second sum is over the delivery delays (normalized to the time deadline T) experienced by the content fractions.

The quantity $\phi_{l,c}^k$ can be computed by evaluating the amount of data that can be transferred at step k (i.e., the flow) over the edges of the type (v_m^k,v_l^k) and (r_i^k,v_l^k), with v_m and r_i being, respectively, a relay and an RSU storing at step k part of, or all, content c requested by v_l. More specifically, for each k, we define the expected flow for content c that is carried over the link associated to a V2V (resp. I2V) contact as $f_c(v_m^k,v_l^k)$ (resp. $f_c(r_i^k,v_l^k)$). From our definitions in Sec. 3.4.1, we have

$$f_c(v_m^k,v_l^k)\le w(v_m^k,v_l^k),\quad f_c(r_i^k,v_l^k)\le w(r_i^k,v_l^k). \tag{3.2}$$

By leveraging the flow definition above, we can write:

$$\phi_{l,c}^k=\frac{1}{s_c}\left[\sum_{(v_m^k,v_l^k)\in\mathcal{E}_v^k}f_c(v_m^k,v_l^k)+\sum_{(r_i^i,v_l^k)\in\mathcal{E}_r^k}f_c(r_i^k,v_l^k)\right],$$

where s_c is the content size.

The evaluation of the expected flows must account for the channel contention among network nodes as well as among flows related to different content transfers. Thus, beside ensuring non-negative flows in the TEG-WP, we need to introduce the constraints listed below.

Flow conservation. For each vertex in the TEG-WP, we impose that the total flow for a content on outgoing edges, scaled by the probability that the corresponding contacts occur, is equal to the total incoming flow for the same content. E.g., in the case of a relay vertex, we have:

$$\sum_{(r_i^k,v_l^k)\in\mathcal{E}_r^k}f_c(r_i^k,v_l^k)=\sum_{(v_l^k,v_m^k)\in\mathcal{E}_v^k}\frac{f_c(v_l^k,v_m^k)}{p(v_l^k,v_m^k)}+f_c(v_l^k,v_l^{k+1})\,. \tag{3.3}$$

As an example, consider the 2-step evolution in Fig. 3.2, where v_m is a downloader for content c. Note that the transmissions from r_i to v_l and from v_l to v_m take place

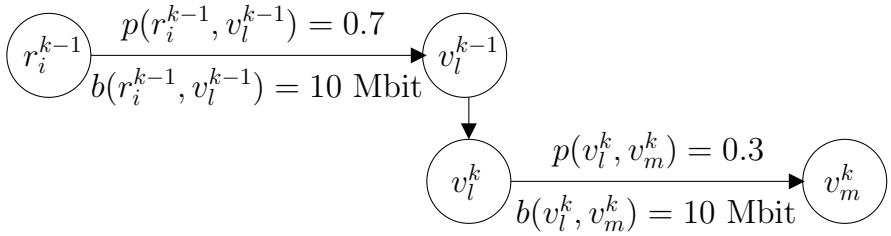


Figure 3.2. Flow conservation: an example

at different steps, thus channel access has no effect here. Intuitively, we can try to transfer 10 Mbit from r_i to v_m, and we will succeed with probability $0.7 \cdot 0.3 = 0.21$. Then, the overall expected flow delivered to the downloader is $0.21 \cdot 10 = 2.1$ Mbit. However, if only the constraints in (3.2) were applied on each of the two intra-step edges, the expected flow should not exceed b times the edge probability. Hence, we could incorrectly conclude that the expected flow from r_i to v_m is $\min\{0.7 \cdot 10, 0.3 \cdot 10\} = 3$ Mbit. Instead, imposing (3.3) for vertices v_l^{k-1} and v_l^k, it correctly results that $f_c(r_i^{k-1}, v_l^{k-1}) = f_c(v_l^k, v_m^k)/p(v_l^k, v_m^k)$, i.e., $f_c(v_l^k, v_m^k) = 2.1$, which is consistent with our intuition.

Flow causality. In order for a node (be it a vehicle or an RSU) to transmit some data (of any content) at step k, such data must have been already downloaded from some other node at step $h \leq k$. In other words, we need to introduce a *causality* constraint, imposing that, at each step k, the data downloaded by node v_m from node v_l until k (as opposed to "during step k alone") is no more than the data v_l obtained until k from other nodes. Thus, for any edge of type (v_l^k, v_m^k) and content c, we have that:

$$\sum_{h=1}^{k} \frac{f_c(v_l^k, v_m^k)}{p(v_l^k, v_m^k)} \leq \sum_{h=1}^{k} \left[\sum_{v_n^h \in \mathcal{V}^h \setminus v_m^h} f_c(v_n^k, v_l^k) + \sum_{r_i^h \in \mathcal{R}^h} f_c(r_i^h, v_l^h) \right].$$

Channel access. We assume that the nodes access the channel using an IEEE 802.11-based scheme with RTS/CTS handshake. Thus, when v_l transmits to v_m, all neighbors of v_l and v_m must be silent. Also, recall that we assume V2V and I2V traffic not to interfere. Then, the channel access constraint for any v_l at step k is:

$$\sum_{\substack{(v_n^k, v_o^k) \in \mathcal{E}_v^k \\ c \in \mathcal{C}}} \mathbb{1}_{[v_n^k, v_l^k]} \frac{f_c(v_n^k, v_o^k)}{b(v_n^k, v_o^k)} + \sum_{\substack{(v_p^k, v_o^k) \in \mathcal{E}_v^k \\ c \in \mathcal{C}}} \mathbb{1}_{[v_o^k, v_l^k]} \left(1 - \mathbb{1}_{[v_p^k, v_l^k]}\right) \cdot$$
$$\frac{f_c(v_p^k, v_o^k)}{b(v_p^k, v_o^k)} + \sum_{\substack{(r_i^k, v_o^k) \in \mathcal{E}_r^k \\ c \in \mathcal{C}}} \mathbb{1}_{[r_i^k, v_l^k]} \frac{f_c(r_i^k, v_o^k)}{b(r_i^k, v_o^k)} \leq 1,$$

where the indicator function is equal to 1 if the specified vertices either are neighbors or coincide, and it is 0 otherwise. The three sums on the left hand side of the inequality account for the fact that the following events cannot take place at the same time: (i) v_l or a vehicle within range of v_l transmit, (ii) v_l or a vehicle within range of v_l receive, (iii) an RSU that is a neighbor of v_l transmits.

As far as RSUs are concerned, we still have to impose that no RSU r_i can transmit for longer than one step:

$$\sum_{(r_i^k,v_l^k)\in\mathcal{E}_r^k} \sum_{c\in\mathcal{C}} \frac{f_c(r_i^k,v_l^k)}{b(r_i^k,v_l^k)} \leq 1\,.$$

In conclusion, at each time step, each RSU r_i formulates an optimization problem aimed at maximizing (3.1) under the above constraints. The solution of the problem yields the optimal prefetching and scheduling decisions, based on the prediction $\mathcal{P}_i(u,H)$. Since all constraints are linear expressions with respect to the control variables f_c's, the problem falls in the LP category, hence it can be efficiently solved in real-time.

3.5 Content delivery through V2V relaying

When the solution of the LP problem leads an RSU to schedule transmissions to relays, the latter are in charge of delivering the data to the downloaders. We envision two different approaches to manage V2V data relaying, as detailed next.

RSU-driven relaying. The solution to the optimization problem formulated by each RSU, as described in Sec. 3.4.2, implicitly schedules relay-to-downloader data transfers in addition to RSU-to-downloader and RSU-to-relay ones. Such a scheduling is optimal with respect to the contact prediction available at each RSU and the requests it is aware of, and it can be easily leveraged to drive V2V transfers. To that end, it is sufficient that, based on the contacts they foresee, RSUs provide the relay vehicles with the identity of the downloaders the data are intended for, as well as the expected contact times. Relays will then use this information to decide on when to establish a V2V connection with a given downloader.

Since this approach is equivalent to employing the LP problem solution computed by each RSU to take decisions not only on I2V transfers, but also on V2V transmissions, we expect its performance to be highly dependent on the prediction accuracy. More precisely, uncertainty in the contact estimation can lead either to failure in delivering the data, if a foreseen V2V link turns out not to be established, or to a waste of opportunities, if an exploitable V2V contact is not predicted. Additionally, the scheduling computed by different RSUs may result to be incompatible, since they are generated from different TEG-WPs: this leads to unexpected channel contention and consequent

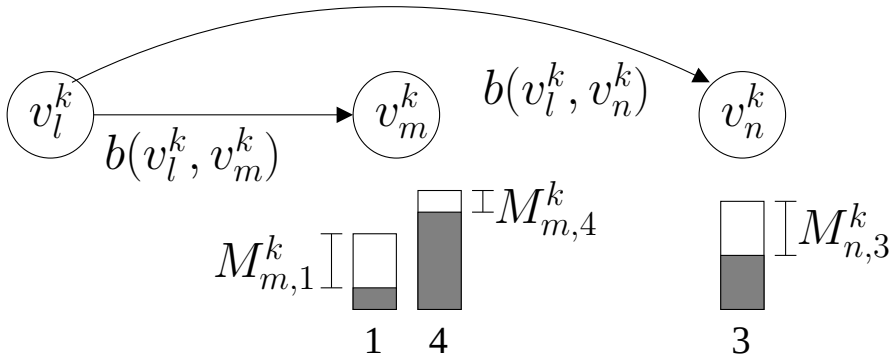


Figure 3.3. Greedy relaying example. In phase 1, downloaders v_m and v_n have incomplete contents 1, 4 and 3, respectively, and announce the missing data. In phase 2, relay v_l, storing all missing data, allocates its airtime to satisfy the requests by v_m and v_n, adopting a water-filling approach

delays, or impossibility to deliver all data. Note also that the scheduling at RSUs does not account for content requests issued after the last update received from the traffic manager.

Greedy relaying. A dual approach to the RSU-driven relaying consists in letting V2V transfers take place in a greedy fashion, by exploiting any opportunity to make incomplete downloads progress. In this case, the LP problem is only employed to take prefetching and I2V transfer decisions at the RSUs, while relays and downloaders autonomously manage V2V transfers. The greedy relaying protocol we adopt involves three phases and is repeated periodically.

In the first phase, each downloader advertises the list of contents it is currently downloading, detailing, for each of them, the amount of data it needs to complete the transfer. As shown in Fig. 3.3, a generic downloader v_m will thus announce, at step k and for each incomplete content c, the quantity $M_{m,c}^k = s_c \cdot \left(1 - \sum_{i=t_{m,c}}^{k-1} \phi_{m,c}^i\right)$. The missing data information broadcast by downloaders is received by relays within range. Note that this phase requires loose synchronization (with accuracy of the order of 1 ms) among nearby vehicles, which can be easily obtained through, e.g., GPS, and is already foreseen in the current standards for vehicular networks.

In the second phase, each relay filters the missing data requests received from downloaders in its neighborhood, only retaining those for contents it actually stores. Then, based on the SNR computed on the received broadcast transmission, it estimates the link data rate b, hence the time needed to complete each of the retained transfers. For instance, in Fig. 3.3, the time computed by relay v_l to complete the transfer to downloader v_m of a content c is $T_{m,c}^k = M_{m,c}^k / b(v_l^k, v_m^k)$.

A relay then decides how to serve the requests, by formulating and solving a max-min fairness problem. The rationale behind such a choice is that a max-min fair allocation of the airtime allows downloads to progress evenly, not favoring large downloads over small ones or vice-versa, yet guaranteeing that the medium is fully exploited.

Denoting the total airtime to be used for data transfer by Δ, the relay assigns a portion of time $0 \leq t^k_{m,c} \leq \Delta$ to each downloader, such that the resulting allocation $\mathbb{T} = \{t^k_{m,c}\}$ solves the problem:

$$\max \min_{\mathbb{T}} \left(\mathbb{1}_{\left[t^k_{m,c} < T^k_{m,c}\right]} t^k_{m,c} \right), \quad s.t. \sum_{t^k_{m,c} \in \mathbb{T}} t^k_{m,c} \leq \Delta \,. \tag{3.4}$$

A water-filling approach is employed to efficiently solve (3.4). Once the locally-optimal allocation is obtained, in the third phase (of duration equal to Δ) relays start to transmit their data to target downloaders. If multiple relays are neighbors, or hidden terminals to each other, their allocations will have to share the medium according to the constraints on channel access defined earlier in this section.

3.6 Results

Here, we detail the mobility and communication scenario we take as a reference and present the impact of the parameters of the fog-of-war model on the contact prediction. The results on content downloading in the reference scenario follow.

3.6.1 Reference scenario

We consider a real-world road topology representing a 3×3 km^2 section of the urban area of Turin, Italy, portrayed in Fig. 3.4. We focus on 30 minutes of consistently fluid traffic conditions [35], such that, at any instant, the scenario includes about one thousand vehicles simultaneously traveling over the area and taking part in the ITS. The vehicular mobility has been synthetically generated using the SUMO simulator. The time granularity of the resulting mobility trace is 1 s, hence we set the granularity of the traffic manager prediction and the periodicity of the execution of the V2V data relaying protocol to 1 s.

Fig. 3.4 also depicts the default deployment that we assume for the ITS infrastructure, with 10 RSUs located at the most crowded intersections, represented by green dots. Based on the findings in [23], such a placement strategy allows ITS-based downloading to perform close to the optimum.

With regard to the communication technology, we assume that, like vehicles, RSUs have one DSRC interface only: the extension to the case where they have more than one interface is straightforward. At the physical layer, RSUs operate on the same frequency

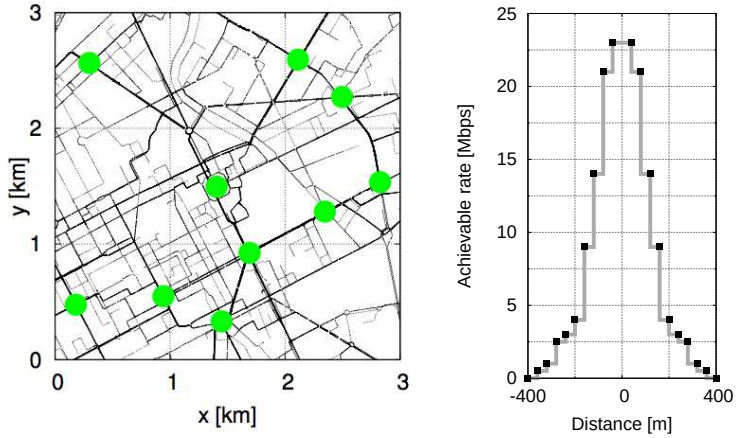


Figure 3.4. Road topology (left) and achievable network-layer rate (right)

channel used for V2V communications. At the MAC layer, the available bandwidth in the ITS is shared by the nodes using an 802.11-based protocol, with RTS/CTS handshake. We assume that rate adaptation is employed, hence the value of the achievable network-layer rate between any two nodes is set according to their distance. In particular, we refer to the 802.11a experimental results in [4, Fig. 5] to derive the values shown in Fig. 3.4, and we use them as samples of the achievable network-layer rate. Also, we limit the maximum radio range of any node to 200 m, since, as stated in [4], this distance allows the establishment of a reliable communication in 80% of the cases.

As for the cellular network, we assume that full cellular coverage of the area is available. A user can always complete its download through the cellular infrastructure if it could not retrieve the whole content through the ITS within T seconds. Unless otherwise specified, we set T=120 s.

Users' content demand is modeled by assuming that $|\mathcal{C}|$=100 items are available and have the same size s_c=10 MBytes. The per-user request rate is Poisson distributed with rate $\lambda = 0.005$. When social groups are considered, we represent them as vehicular flows, reflecting, e.g., the case of users traveling toward a business district and wishing to download financial news clips or market updates. In order to identify the vehicular flows, at every time step we run the κ-means clustering algorithm on the mobility trace. Then, we consider clusters, detected in consecutive steps and having the closest centroids, to be snapshots of the same flow. In particular, we use $\kappa = 5$ so as to track the 5 largest social groups over time (each group turns out to include at least 10 vehicles).

Finally, we assume that the traffic manager generates its predictions every 30 seconds, forecasting the next 30 seconds of contacts. Since δ=1 s, this implies H=30 in the following.

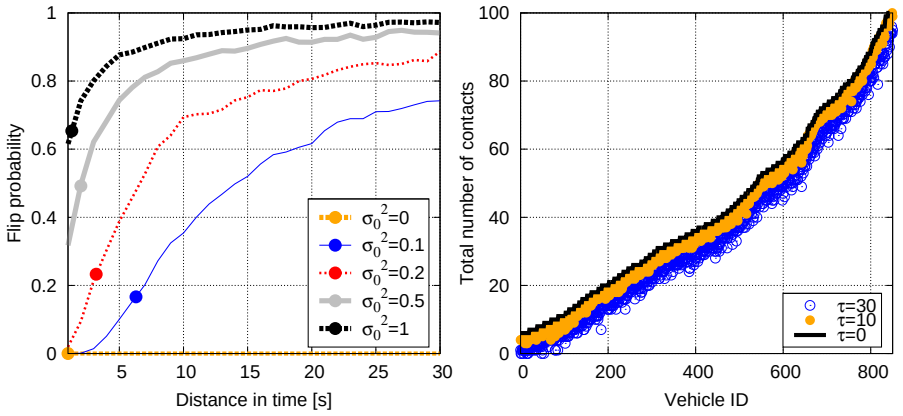


Figure 3.5. Left: contact flip probability vs. the prediction time-span, for $\tau = 0$ and varying σ_0^2. Dots represent the average probability value. Right: number of contacts for each vehicle in the mobility trace, for $\sigma_0^2 = 0$ and varying τ

3.6.2 Behavior of the fog-of-war model

The impact of the fog-of-war model parameters, in the above reference scenario, is shown in Fig. 3.5.

The left plot presents the probability of contact flip, i.e., that an actual contact is removed and a spurious one is created, as a function of the time before the contact is scheduled to begin (i.e., $k-u$, u being the step at which the prediction is compiled). The curves are obtained for $\tau = \rho = 0$ and different values of σ_0^2, with $\sigma_0^2 = 0$ corresponding to a flawless prediction. As expected, the larger the σ_0^2, the higher the probability to predict spurious contacts. Also, the time span $k-u$ has a significant impact, as contacts established farther in the future become less predictable and are affected by a higher flip probability.

In the plot, the dots on the curves represent the flip probability computed over all actual contacts observed within the prediction horizon. Note that for $\sigma_0^2 \geq 0.5$ the majority of predicted contacts are spurious, while for $\sigma_0^2 = 0.1$ we have a quite reliable prediction (about 4 out of 5 actual contacts are correctly forecast). This is due to the fact that contacts already existing at step u are associated with a null distance in time, hence they are always correctly predicted.

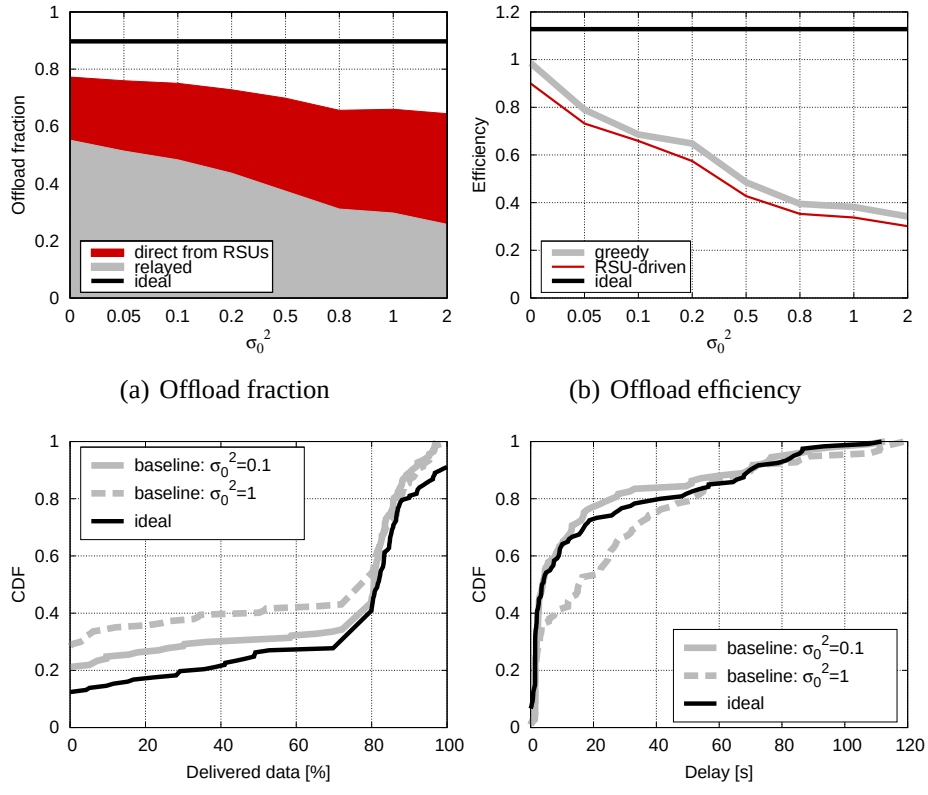
The right plot shows instead the impact of τ, when $\sigma_0^2 = \rho = 0$. More precisely, we report the total number of contacts per vehicle, over the vehicle's trip, as τ varies; clearly, $\tau = 0$ corresponds to the actual contact duration statistics. On the x axis, the vehicles are ordered according to the increasing number of actual contacts they have. Note that the larger the τ, the higher the probability that contacts are evicted and do not appear in the prediction at all. The impact of τ is especially evident for the vehicles with a total number of contacts below 60, for which the shorter contacts that tend to be evicted represent a significant percentage.

Results showing the impact of ρ on the data rate at which nodes communicate are omitted because of both lack of space and the marginal impact that this parameter turned out to have on the system performance (see Sec. 3.6.3).

3.6.3 Performance of content downloading

We evaluate the effectiveness of offloading content download from cellular to ITS networks, in the reference scenario previously described.

We first assume (i) a content demand process where each content is requested by vehicles with equal probability, (ii) unlimited time validity for contents, and (iii) $B = 100$ Mbit/s, i.e., high-bandwidth links connecting the RSUs with the content servers. Note that this essentially implies ideal ITS operation, as RSUs need to download contents only once, thanks to their unlimited cache size and the infinite content validity. We refer to this scenario as our *baseline* system configuration, and employ it to study the impact on the download performance of: forecast accuracy, V2V relaying strategy,



(a) Offload fraction

(b) Offload efficiency

(c) Offload distribution

(d) Delay

Figure 3.6. Content download performance in presence of cellular network offloading via ITS-based communication, under the baseline system configuration

deadline for the ITS data retrieval and ITS infrastructure dimensioning. The rationale is that the baseline scenario allows us to evaluate the performance of the wireless portion of the system, while avoiding biases due to the demand distribution or to backbone limitations.

As a second step, we relax the assumptions on the RSU content retrieval operation, content demand and validity, and investigate: (i) a *constrained* system configuration, where the content validity is limited in time and the RSU backbone bandwidth is reduced; (ii) a *social* system configuration, where the content requested by vehicles is influenced by the traffic flow they belong to. The latter configuration also allows us to compare the offloading performance obtained with a contact prediction with that achieved by a forecast-agnostic, push-based scheduling scheme based on content popularity.

Baseline scenario. The performance of the offloading process in the baseline scenario is presented in Fig. 3.6(a), which portrays the average fraction of requested content that a vehicle can successfully download through the ITS before the expiration of the deadline T. The results have been obtained as σ_0^2 varies, under the greedy relaying scheme and for $\tau = \rho = 0$.

The offload fraction is broken down into content retrieved directly from RSUs and content obtained from relays through V2V communication, and it is compared against the ideal offload performance. The latter is derived by solving the optimization problem for $\sigma_0^2 = 0$, a very large prediction horizon (namely, $H = 300$) and assuming that future user requests are known a priori; this enables perfect I2V scheduling.

Firstly, we observe that ITS can relieve the cellular network of 70-80% of the cost associated to content download. Secondly, a great part of the merit goes to V2V relaying, bearing between 30 and 60% of the content transfer effort, which confirms that opportunistic transfers are highly beneficial in the offload process. Thirdly, the overall performance is not too far from the ideal one, which would allow a 90% offload.

The impact of the accuracy of the contact prediction is shown by varying σ_0^2. Quite surprisingly, very accurate predictions (low values on the x axis) result in a performance that is just slightly better than that scored by almost random contact estimations (high σ_0^2's). Inaccurate predictions lead however to a reduced contribution of V2V with respect to I2V transfers, as the former drops from more than 75% to less than 40% of the overall offloaded fraction.

The actual cost of an imprecise contact prediction is revealed by Fig. 3.6(b), which shows the offload efficiency, i.e., the ratio of the amount of data delivered to a downloader to that transmitted by the RSUs (to either downloaders or relays). A low efficiency implies a waste of wireless resources at the RSUs, while a high efficiency means that only useful ITS-based transfers are performed. The efficiency can be higher than one, since a relay can download some content (or part of it) and then provide it to multiple downloaders. The plot clearly shows that, in order to maintain high offload

fractions, the less precise the information on future contacts, the larger the amount of data the RSUs have to transfer to relays.

Another interesting fact underscored by Fig. 3.6(b) is that RSU-driven relaying is outperformed by the greedy approach. The reason for such a behavior is that the amount of data transmitted by the RSUs is the same in the two cases, but the former is unable to exploit data transfers to future downloaders (of which RSUs are unaware). This is an important contribution to the performance, unlike the optimized RSU-driven scheduling that is beneficial only in the rare case of multiple, simultaneous relay-downloader transfers. As a consequence, the greedy approach is to be preferred and we will focus only on it in the following.

Fig. 3.6(c) further details the offload performance, showing the cumulative distribution function (CDF) of the fraction of content that each downloader can retrieve through the ITS. Results are shown for quite accurate ($\sigma_0^2 = 0.1$) and rather imprecise ($\sigma_0^2 = 1$) predictions, and benchmarked against the ideal case. The CDFs clearly identify two larger classes of downloaders: those that can get a very small percentage (possibly zero) of the data they request, and those (over 50% of the total) who can obtain almost all (80% or more) of the data through ITS. Interestingly, the latter category does not seem to be affected by σ_0^2, as the curves are very close for high values on the x axis. On the contrary, the percentage of downloaders unable to get any data is sensibly reduced as the contact estimation precision grows. We can thus conclude that an accurate prediction is most useful to offload downloads for users that are hard to reach, e.g., because traveling on secondary roads.

Finally, Fig. 3.6(d) portrays the CDF of the delay in the ITS-based content delivery. A large portion of the data, amounting to 70% of the content size, is typically obtained within a short timespan (approximately 20 s). The results are similar in presence of ideal and precise contact predictions, although in the ideal case the higher fraction of downloaded contents leads to an increased latency for users on unfavorable routes. An inaccurate contact prediction, instead, yields quite higher delays.

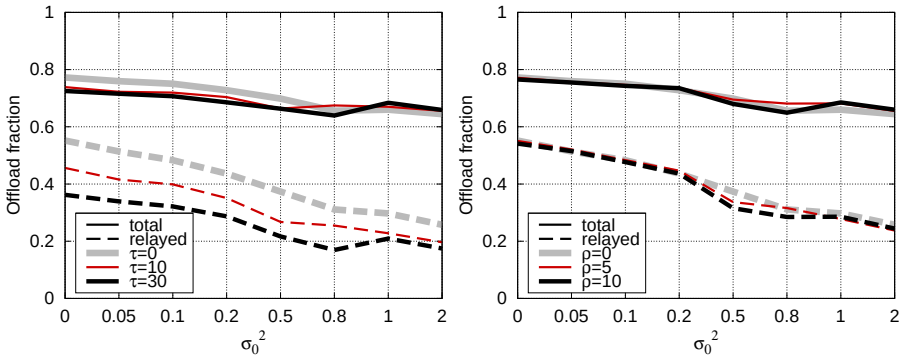


Figure 3.7. Offload fraction as τ (left) and ρ (right) vary

Next, we evaluate the impact of the prediction parameters τ and ρ, when $\sigma_0^2 = 0$. The left plot in Fig. 3.7 highlights that the larger the τ, the less V2V transfers can be exploited. Indeed, for high values of τ the number of predicted contacts decreases, as shown in Fig. 3.5; additionally, the number of contacts with short predicted duration, which tend not to be used, increases. The impact of ρ is instead marginal, as evident from the right plot in Fig. 3.7. The results thus suggest that it is important to accurately predict the contacts and their duration, but not their data rate. In the following, we set $\tau = \rho = 0$.

Tab. 3.1 shows the offload fraction for varying σ_0^2 and number of deployed RSUs. As expected, increasing the number of RSUs favors the ITS-based offloading process. However, improving future contacts estimation can compensate for a less pervasive ITS coverage. Indeed, by cross-checking similar offload fractions over different columns, we note that an accurate prediction requires between 20 and 30% less RSUs, while maintaining similar performance.

The benefits of an accurate prediction are also shown in Tab. 3.2, which reports the offload fraction for different values of σ_0^2 and T (the time after which users start retrieving data from the cellular network). Indeed, the higher the T, the larger the amount of data downloaded through the ITS, however improving the forecast reliability pays significantly more than delaying the use of the cellular network.

Summary: Our results show that ITS is a viable alternative, or complementary solution, to cellular networks for content downloading by mobile users. In particular, if a relatively reliable mobility prediction is available, the offload of the cellular infrastructure can be achieved by sparing wireless resources, better serving downloaders on secondary roads, reducing the download latency, and lowering the ITS deployment cost.

Constrained scenario. Here, we focus on the case of RSU backbone links with bandwidth limited to $B = 10$ Mbps and contents expiring after an exponentially distributed time with mean equal to 200 s. We remark that the latter condition forces, upon expiration of a content, both RSUs and downloaders to discard any portion of the content they previously obtained, and restart the download from scratch.

The offload fraction obtained in such a constrained configuration is presented in Fig. 3.8, where it is compared with our baseline. More precisely, the left plot shows the average offloading fractions as σ_0^2 varies, while the right one details the per-downloader CDF of the offload fraction. The first plot clearly evidences that the introduction of the

Table 3.1. Offload fraction as the number of RSUs and σ_0^2 vary

σ_0^2 \ No. RSUs	6	8	10	12	14	16
0.1	0.55	0.67	0.76	0.79	0.92	0.94
1	0.48	0.57	0.66	0.71	0.82	0.84

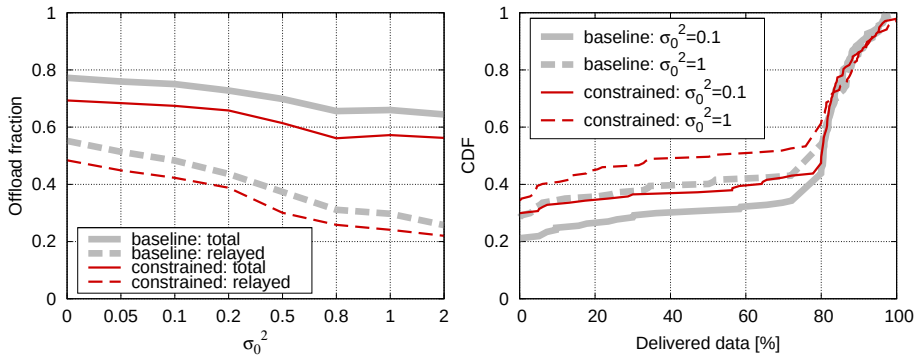


Figure 3.8. ITS-based download performance in the constrained configuration

constraints leads to an unchanged trend with respect to the contact prediction accuracy, at the cost of a remarkable performance reduction. Interestingly, the performance drop mainly concerns the download via V2V relaying, since, upon expiration of the content, relays have to discard the data and cannot help in their delivery any longer. In the second plot, we can once more observe how less performing network operations affect downloaders on unfavorable routes (e.g., traveling on secondary roads).

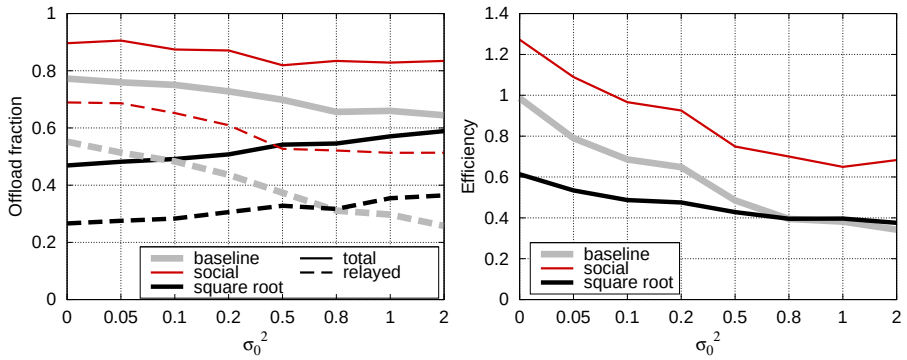


Figure 3.9. Impact of social groups on ITS-based download performance and comparison against a content popularity-based approach

Social scenario. We now evaluate the offload performance in presence of users belonging to social groups. Such users request contents according to a Zipf's distribution with

Table 3.2. Offload fraction as the deadline T and σ_0^2 vary

σ_0^2 \ T [s]	60	120	180	240
0.1	0.69	0.75	0.78	0.80
1	0.58	0.66	0.71	0.72

exponent equal to 2; distributions related to different groups are shifted by 20 contents with respect to each other. Vehicles not belonging to any group request contents with equal probability.

Fig. 3.9 shows that, in presence of social groups, the amount of data the downloaders can retrieve through the ITS significantly increases, mostly due to V2V relaying. Indeed, thanks to the tighter correlation between the vehicles' routes and the content they request, it is likely that the desired information is obtained from nearby vehicles. This also explains the very high efficiency of relayed traffic in the right plot of Fig. 3.9.

The plots also portray the performance of an approach based on content popularity that does not exploit the mobility forecast but assumes knowledge of the content popularity distribution. Specifically, it lets RSUs select the content to be pushed towards a relay, with a probability proportional to the square root of the content popularity [36]. For fair comparison, we force the amount of data sent by the RSUs to match what is observed in our prediction-based scheme with social groups.

The results show that predicting the contacts allows for significantly better performance than the knowledge on the content popularity. This is due to the high mobility of our scenario: either the content is delivered to the right social group, or retrieving the content from a vehicle carrying the data becomes very hard. Such an observation is confirmed by the curve referring to relayed traffic in the left plot, which is significantly lower for the square root approach than in the predicted-based scheme. As a last remark, the offload fraction in the square root case grows with the increase of σ_0^2, since more data are injected in the network to match the amount observed in the social case. Nevertheless, such an increase in the delivered data does not make up for the higher radio resource consumption, thus leading to a lower efficiency.

3.7 Related work

A few works have studied scenarios where ITS and cellular technologies coexist so as to allow vehicular users to download contents from the wired network. However, their scope and methodology differ from ours. In particular, in [16], only I2V direct transfers are considered, and the focus is on the prefetching of contents at RSUs, which are assumed to have high-latency, low-bandwidth links to the Internet. The objective is then to optimize the usage of such links, by leveraging estimates of the amount of traffic the vehicles will be able to download from each RSU. Moreover, although the system in [16] comprises cellular coverage, its use is limited to signaling purposes. Conversely, the works in [37, 38] investigate to which extent DSRC-based vehicular contacts can help to offload the cellular network, in a scenario that does not include RSUs and where the same content must be disseminated to all users within a delay threshold. The problem is then to determine how many copies of the content shall be injected in the network and which vehicles are most suitable to receive them. Note

that, unlike previous works, we study content downloading in ITS accounting for all communication methodologies, i.e., I2V, V2V, and cellular-based, at a time. This allows us to jointly investigate the problems of content prefetching at RSUs, scheduling of I2V transfers, and management of opportunistic V2V transfers.

The approach we adopt relates our work to the problem of transmission scheduling in wireless networks, which has been widely studied. However, most works address the case of connected multi-hop networks, e.g., [39], or social delay-tolerant networks, e.g., [40]. The vehicular environment mixes elements of both, thus solutions that assume full reachability or contacts periodicity [41] in the order of hours or days do not apply to our context. A scheduling and prefetching scheme for content downloading explicitly designed for vehicular networks is presented in [42]. This work, however, employs simple road topologies and simplistic mobility models, and does not consider the presence of a cellular infrastructure. As further additions to the literature on transmission scheduling to vehicles, we take into account, for the first time, the role that mobility-based communities have in the generation of content demand, and evaluate the impact of uncertainty in the estimation of future I2V and V2V contacts.

Concerning the latter aspect, there are several ongoing efforts on inferring future vehicular contacts, given the current position and past car trajectories [31, 32, 43]. Thanks to our fog-of-war model, our system can use any of these techniques, including future, more accurate ones, as an input.

Finally, the representation of a time-varying network as a time-expanded graph has also been employed in [23, 24]. Besides the different scope, the time-expanded graphs we propose differs from the above representations as we introduce probabilistic edge weights, in order to model uncertainty in the prediction of inter-node contacts.

3.8 Conclusion

Congestion of cellular infrastructure caused by growing data traffic can be addressed either by investing in backhaul provisioning or by finding alternative solutions for content delivery to mobile users. In our work, we investigated the latter approach for the support of content downloading in a vehicular environment. The issue of content prefetching and data transmissions scheduling from roadside units was analyzed in the realistic case of finite-horizon, inaccurate mobility prediction. We showed that if the prediction error is not overwhelming, vehicles can be effectively served by the ITS, either through direct download from RSUs or by relaying, thus relieving the cellular networks from the download traffic. The offload efficiency we obtained was close to an ideal case and significantly better than that of a content popularity-based solution. Further benefits can be garnered in presence of identifiable social groups among vehicular users, whose interest affinity can be leveraged to deliver the right content to the right user through nearby vehicles.

Chapter 4

Infrastructure deployment in a non-cooperative setting

4.1 Introduction

Unlike previous chapters and most works in literature [1,2,9–12], where the RSU infrastructure is owned by a single operator, in this chapter we aim at studying the dynamics of scenarios where different operators may competitively deploy their RSUs to attract the largest number of customers.

Without purporting to provide a comprehensive solution, (i) we set the problem of RSU deployment that maximizes the revenue for a content provider within the framework of game theory; (ii) we derive preliminary results, that can be extended for a more general approach to the problem and (iii) we verify the validity of our approach through simulation.

4.2 Reference Scenario Description

We consider a scenario with two operators O_1 and O_2, which would like to deploy Road Side Units (RSUs) for distributing content along a stretch of road of length D. Each operator can deploy its RSUs at a subset of a set of candidate sites J. Each RSU is characterized by a coverage range R, which defines its service area, and by an application-level goodput c for content delivery. The goodput depends on the wireless technology the RSU is equipped with and on the communication protocols used for content delivery. Furthermore, in practice the goodput might be affected by physical layer impairments, interference, and collision with other transmissions to/from the same or different RSUs.

We account for such impairments in the form of inter-RSU interference. In the considered model the inter-RSU interference is a function of the distance d between the

interfering RSUs, and it determines the goodput that the individual RSUs can achieve. We express the goodput of each RSU at distance d as

$$c(d) = \frac{c}{1 + \omega(d)}, \tag{4.1}$$

where the interference function $\omega(d)$ is a monotone non-increasing function of d. The interference function is bounded, $\Omega = \omega(0) \geq \omega(d) \geq \omega(D) \geq 0$ for $0 \leq d \leq D$. The assumption of monotone non-increasingness is rather reasonable, since the interference level usually does not increase when interferers move away. Furthermore, we consider the relevant case when $\Omega > 1$. As an example, if two RSUs are deployed at the same candidate location then they could use a MAC protocol to share the physical medium, and their total capacity would be $\frac{2c}{1+\Omega} < c$.

There is a bidirectional flow of vehicles on the considered stretch of road; λ_A is the intensity of the flow of vehicles from left to right, and λ_B is the intensity from right to left. The vehicles move at some constant speed v[m/s]. Each vehicle aims to retrieve some content with an average size of S[bits], Depending on the content size, multiple MAC layer frames may be required to accomplish the content download. Content retrieval is attempted from the first met RSU along the road upon completion of a standard association procedure. In case multiple RSUs (from different operators) are available simultaneously, the RSU association is done at random. A content download is successful if the vehicle manages to retrieve the content before leaving the coverage area of the RSU. If the content retrieval is unsuccesful, the vehicle attempts to download the content via the next RSU encountered along the road. We define the offered load as $\rho_A = \lambda_A S$ and $\rho_B = \lambda_B S$ in the two directions, respectively. This definition of load does not consider factors such as the number of vehicles in the coverage area, the content size, or the ratio of successful content retrievals, but it is appropriate for our purposes. We consider that the revenue of an operator in a deployment is proportional to the traffic load it serves, that is, to the number of vehicles that successfully get service through the operator's RSU. Figure 4.1 shows a scenario with two candidate sites for RSU deployment ($J = \{A,B\}$) at the two extremes of a stretch of road.

4.3 RSU Deployment Optimization

We start off by considering a scenario where one operator deploys its fixed RSU at the beginning (left side) of the stretch of road. The goal of the other operator is then to install its non-fixed RSU at a distance $d \in [0,D]$ such as to maximize its own utility. Without loss of generality, we assume that the offered load $\rho_A \geq \rho_B$. Our goal is to characterize the best response of the second operator, i.e., the optimal distance d to deploy the RSU.

The utility function of the non-fixed RSU depends on the offered loads in the two directions, on the RSUs' goodput as a function of d, and on the spillover traffic of the

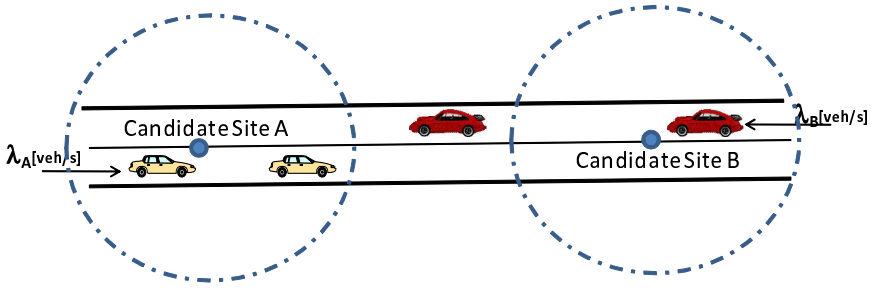


Figure 4.1. Reference scenario with two candidate sites at the two extremes of the stretch of road.

fixed RSU. The spillover $\rho^s(d)$ is the part of the offered load that exceeds the capacity of the fixed RSU, and hence is not served. That is, $\rho^s(d) = max(0,\rho_A - \frac{c}{1+\omega(d)})$. Clearly, if $\rho_A < \frac{c}{1+\Omega}$ then there is no spillover. Otherwise, there is a distance d_s between the RSUs such that for $d < d_s$ the spillover is strictly positive ($\rho^s(d) > 0$), and for $d > d_s$ there is no spillover ($\rho^s(d) = 0$). At distance d_s the interference-limited capacity of the fixed RSU, $c\,(d_s)$, is equal to the offered load ρ_A. Therefore, we have:

$$d_s = \omega^{-1}\left(\frac{c}{\rho_A} - 1\right), \tag{4.2}$$

where $\omega^{-1}(\cdot)$ is the inverse function of the interference function.

In order to analyze the utility function of the second operator we consider two functions. The *traffic function*, $t(d) = \rho^s(d) + \rho_B$, expresses the traffic offered to the non-fixed RSU, and is a monotone non-increasing function of d. Note that the spillover $\rho^s(d)$ makes $t(d)$ depend on distance d. In particular:

$$t(d) = \begin{cases} \frac{\rho_A+\rho_B}{2} & \text{if } d = 0 \\ \rho_A + \rho_B - \frac{c}{1+\omega(d)} & \text{if } 0 < d \leq d_s \\ \rho_B & \text{if } d > d_s \end{cases} \tag{4.3}$$

The *capacity function*, $c(d)$, is defined in (4.1). It is a monotone non-decreasing function of d and represents the maximum goodput (thus, utility) the non-fixed RSU can achieve. The utility function $U(\rho_A,\rho_B,d)$ of the non-fixed RSU is the minimum of the two functions. Fig. 4.2 illustrates the three functions for a scenario when $d_s < D$, and hence the utility is constant for $d > d_s$.

According to the value of the total traffic load $\rho_A + \rho_B$, we can identify three operating regimes of the system.

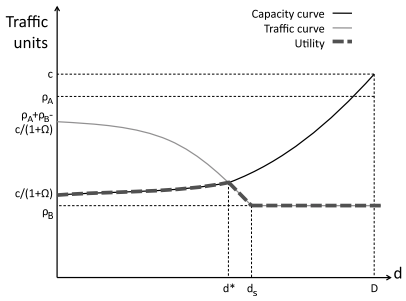


Figure 4.2. Analytic scenario.

High load

In this scenario the offered load exceeds the total maximum capacity of the RSUs. Therefore, the best choice is to place the RSUs at the maximum distance ($d = D$) where there is no interference which could decrease the RSU capacity ($\omega(D) = 0$). This happens when $\rho_A + \rho_B \geq 2c$.

Low load

In this scenario the offered load is so low that the interference effect can be neglected, namely, the traffic is less than the goodput of colocated RSUs. Since $\rho_A > \rho_B$, the non-fixed RSU achieves higher utility if it is colocated and shares $\rho_A + \rho_B$. When it is not colocated, it can only serve ρ_B. This happens when $\rho_A + \rho_B < \frac{2c}{1+\Omega}$

Traffic stealing

In this scenario the interference effect cannot be neglected, but rather, the non-fixed RSU can exploit it to increase its utility. Indeed, by moving closer to the fixed RSU, it can increase the spillover at the fixed RSU and serve more traffic. In other words, the non-fixed RSU has unused capacity and accepts to reduce it if the higher interference allows it to steal some load from the fixed RSU.

It is interesting to investigate the best distance d^* where the utility function of the second operator is maximized. Due to the monotonicity of the traffic and capacity functions the distance d^* that maximizes the utility is the distance at which the two curves intersect in $d^* \in (0,D)$. The two possible intersection points are

$$d_1^* = \omega^{-1}\left(\frac{2c}{\rho_A + \rho_B} - 1\right) \text{ or } d_2^* = \omega^{-1}\left(\frac{c}{\rho_B} - 1\right).$$

However, since the interference function $\omega(d)$ is monotone non-increasing and $\rho_A \geq \rho_B$, we have:

$$\frac{2c}{\rho_A + \rho_B} - 1 \geq \frac{c}{\rho_B} - 1 \geq \frac{c}{\rho_A} - 1,$$

thus

$$\omega^{-1}\left(\frac{2c}{\rho_A + \rho_B} - 1\right) \leq \omega^{-1}\left(\frac{c}{\rho_B} - 1\right) \leq \omega^{-1}\left(\frac{c}{\rho_A} - 1\right).$$

This means that the only intersection point, and thus the maximum utility value, is $d^* = d_1^*$. The optimal distance d^* is no greater than d_s, the distance at which the spillover traffic ρ^s becomes zero. Clearly, the exact value $d^* = \omega^{-1}\left(\frac{2c}{\rho_A+\rho_B} - 1\right)$ depends on the interference and the used technology.

Finally, we investigate under what conditions the optimal distance d^* is in $(0,D)$. Since $\omega(d)$ is monotone, $\omega(0) = \Omega$, and $\omega(D) = 0$, imposing $d^* = \omega^{-1}\left(\frac{2c}{\rho_A+\rho_B} - 1\right)$ to lie in the interval $(0,D)$ implies:

$$0 < \frac{2c}{\rho_A + \rho_B} - 1 < \Omega. \tag{4.4}$$

Therefore, we have that the optimal distance d^* is in $(0,D)$ when $\frac{2c}{\Omega+1} < \rho_A + \rho_B < 2c$. The interval above ($\rho_A + \rho_B \geq 2c$) describes the *High Load* scenario and the interval below ($\rho_A + \rho_B \leq \frac{2c}{\Omega+1}$) describes the *Low Load* scenario. No other case exists.

4.4 RSU Deployment Games

Consider now that both operator can choose where to deploy their RSUs. Since both players aim to maximize their utilities, the problem of RSU deployment can be best modeled as a non-cooperative game. We consider the simple case of two candidate locations at the two extremes of the stretch of road, as shown in Fig. 4.1. To simplify notation, we consider that the inter-candidate site distance is large enough so that the interference between RSUs deployed at different candidate sites is negligible, i.e., $\omega(D) = 0$. Our results can be easily generalized to non-zero interference.

Operator i ($i \in \{1,2\}$) can deploy a single RSU at one of the candidate locations A and B. The goal of each operator is to maximize its own revenue. As we will see, even the simple case of two locations gives rise to a rich set of solutions.

Clearly, the deployment choice of operator O_1 influences the revenue of operator O_2, and vice versa, and the operators' choices influence what portion of the offered

traffic will be served. Let us denote by ρ_A^u (ρ_B^u) the unserved traffic offered by vehicles traveling from left to right (right to left). If the two RSUs are colocated then the operators share a total revenue of $max[\rho_A + \rho_B, \frac{2c}{1+\Omega}]$, and the unserved traffic is $\rho_A^u = \rho_B^u = max[0, (\rho_A + \rho_B - \frac{2c}{1+\Omega})/2]$. Consider now that the RSUs are not colocated, and denote by ρ_A^s (ρ_B^s) the spill-over offered traffic after passing the first RSU location. Then we can write

$$\rho_A^s = max[0, (\frac{\rho_B^s + \rho_A - c}{\rho_A + \rho_B^s})\rho_A], \tag{4.5}$$

$$\rho_B^s = max[0, (\frac{\rho_A^s + \rho_B - c}{\rho_B + \rho_A^s})\rho_B], \tag{4.6}$$

$$\rho_A^u = max[0, (\frac{\rho_A^s + \rho_B - c}{\rho_B + \rho_A^s})\rho_A^s], \tag{4.7}$$

$$\rho_B^u = max[0, (\frac{\rho_B^s + \rho_A - c}{\rho_A + \rho_B^s})\rho_B^s], \tag{4.8}$$

where ρ_A^u(ρ_B^u) is computed considering that the unserved traffic is the spill-over traffic of location A(B) which is not served by location B(A). Furthermore, we have $\rho_A^u = 0 \iff \rho_B^s = 0$ and $\rho_B^u = 0 \iff \rho_A^s = 0$. We can consequently define the socially optimal RSU deployment as the deployment that minimizes the sum of the unserved traffic, i.e., $\rho_A^u + \rho_B^u$.

4.4.1 Simultaneous Deployment

Let us consider first that the two operators make their deployment choices simultaneously, based on the traffic loads ρ_A and ρ_B. We can model the problem as a strategic game and we are interested in the efficiency of the Nash Equilibria (NE) of the game, which is quantified by the Price of Anarchy (PoA), i.e., the ratio of the total revenue in social optimum and the smallest total revenue in any NE. For the case of simultaneous deployment we can state the following.

Proposition 4.4.1. *For the RSU deployment game the price of anarchy is:* $PoA \leq 1+\Omega$.

Proof. In the simple case when the traffic intensity is symmetric, $\rho_A = \rho_B$, the equilibria are easy to obtain. If $\frac{2c}{1+\Omega} \geq \rho_A + \rho_B$ then any deployment is a NE, while colocation is not an equilibrium otherwise. Furthermore, all equilibria are socially optimal, hence $PoA = 1$.

Under asymmetric traffic the number and efficiency of the equilibria depend on the relationship between the offered traffic ρ_A, ρ_B, and the RSU capacity c. For the analysis we can assume without loss of generality that $\rho_A > \rho_B$. For convenience, let us divide the (ρ_A, ρ_B)-space in three partitions, as shown in Fig. 4.3: (1) $\rho_A + \rho_B < \frac{2c}{1+\Omega}$, (2) $\rho_A + \rho_B > 2c$, and (3) $\frac{2c}{1+\Omega} < \rho_A + \rho_B < 2c$. Partition (3) can further be divided

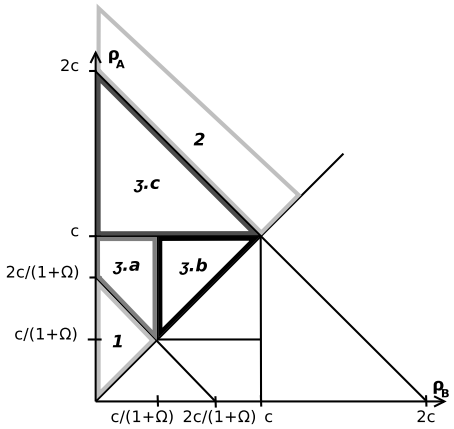


Figure 4.3. (ρ_A,ρ_B)-space partition.

into three partitions. In the following, we analyze the equilibria for the resulting five partitions shown in Fig. 4.3.

Colocation Underload (Partition 1)

The total capacity of the RSUs is higher than the total offered traffic even under colocated deployment, i.e., $\frac{2c}{1+\Omega} > \rho_A + \rho_B$. The NE is (A,A), and the operators have equal revenues. Note that in the equilibrium there is no unserved traffic ($\rho_B^u = 0$, $\rho_A^u = 0$), hence the NE is socially optimal.

Overload (Partition 2)

The offered traffic is higher than the combined RSU capacity, i.e., $2c < \rho_A + \rho_B$. In this case for any $\Omega > 0$ there are two NE, (A,B) and (B,A). To see why, note that in both NE $\rho_B^u > 0$ and $\rho_A^u > 0$. Since both RSUs are fully utilized no player could benefit from colocation. Furthermore, both NE are socially optimal because each operator obtains a revenue c, which is greater than $\frac{c}{1+\Omega}$ with colocation.

Colocation Overload (Partition 3.a)

The total offered traffic exceeds the capacity of colocated RSUs, a colocated RSU can serve one flow entirely, whereas the other flow requires the capacity of a non-colocated RSU, i.e., $\rho_A < c$, $\rho_B < \frac{c}{1+\Omega}$, $\rho_A + \rho_B > \frac{2c}{1+\Omega}$, In this case (A,A) is the unique NE. Observe that under colocation both operators would obtain $\frac{c}{1+\Omega}$ revenue, while under non-colocation one operator would obtain $\rho_B < \frac{c}{1+\Omega}$ (because $\rho_A^s = 0$). In this case the NE is not socially optimal, as non-colocation, which is not a NE, gives a higher revenue,

$\rho_A + \rho_B$. The price of anarchy is

$$PoA = \frac{(\rho_A + \rho_B)(1+\Omega)}{2c} < 1 + \Omega,$$

the inequality holds as $\rho_A + \rho_B < c + \frac{c}{1+\Omega} < 2c$.

Colocation Overload (Partition 3.b)

The offered traffic from both directions exceeds the capacity of colocated RSUs but both flows can be served by non-colocated RSUs, that is, $\frac{c}{1+\Omega} < \rho_B, \rho_A < c$. In this case there are two NE, (A,B) and (B,A). In both NE all traffic is served, hence the NE are socially optimal and $PoA = 1$.

Asymmetric Overload (Partition 3.c)

The traffic from left to right exceeds the RSU capacity but the total offered traffic is less than the total RSU capacity, i.e., $\rho_A > c$ and $\rho_A + \rho_B < 2c$. Table 4.1 shows the operators' revenues for this case.

Table 4.1. Revenue matrix for the case of asymmetric overload (3.c in Fig. 4.3)

		Operator 2 RSU location	
		A	B
Operator 1 RSU location	A	$min(\frac{\rho_A+\rho_B}{2}, \frac{c}{1+\Omega}), min(\frac{\rho_A+\rho_B}{2}, \frac{c}{1+\Omega})$	$c, \rho_A + \rho_B - c$
	B	$\rho_A + \rho_B - c, c$	$min(\frac{\rho_A+\rho_B}{2}, \frac{c}{1+\Omega}), min(\frac{\rho_A+\rho_B}{2}, \frac{c}{1+\Omega})$

The game admits different equilibria depending on the values of ρ_A, ρ_B, and Ω. Following similar arguments as for the previous cases we can state the following.

Lemma 4.4.2. *In the case of asymmetric overload the NE are*

$$\text{NE} = \begin{cases} \{(A,A)\} & \text{if } (\rho_A,\rho_B) \in \textit{Region I} \\ \{(A,B),(B,A)\} & \text{if } (\rho_A,\rho_B) \in \textit{Region II} \end{cases} \tag{4.9}$$

Figure 4.4 illustrates the NE and shows the revenues for the two operators at the equilibrium in partition (3.c). The NE (A,B) and (B,A) are socially optimal, so for the price of anarchy we can state the following.

Lemma 4.4.3. *In the case of asymmetric overload the price of anarchy is*

$$\text{PoA} = \begin{cases} \frac{(\rho_A + \rho_B)(1+\Omega)}{2c} & \text{if } (\rho_A,\rho_B) \in \textit{Region I} \\ 1 & \text{if } (\rho_A,\rho_B) \in \textit{Region II} \end{cases} \tag{4.10}$$

Since $\rho_A + \rho_B < 2c$ we have $\text{PoA} \leq 1 + \Omega$, which concludes the proof. □

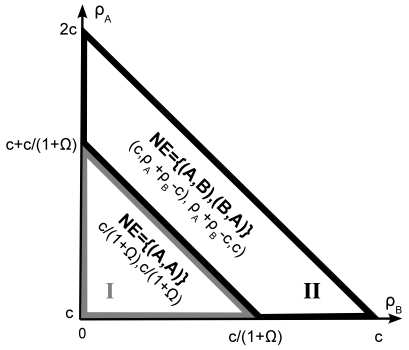


Figure 4.4. NE for the case 3.c: Asymmetric Overload

4.4.2 Leader/Follower Deployment

Let us now consider the scenario where one of the two operators is the market leader and has the first-move advantage. We can model the problem as an extensive-form game and we are interested in its sub-game perfect NE.

The NE derived in Section 4.4.1 for partitions 1, 2, and 3.a in Fig. 4.3 can be easily shown to be sub-game perfect. Nevertheless, not all NE in partitions 3.b and 3.c are sub-game perfect.

Colocation Overload (Partition 3.b)

From the two NE (A,B) and (B,A) only (A,B) is sub-game perfect. Indeed, the two NE have revenues (ρ_A,ρ_B) and (ρ_B,ρ_A), respectively. Since, $\rho_A > \rho_B$, operator O_1 will deploy its RSU in A, thus, the best choice for operator O_2 will be to choose B, and hence the revenue for the first-mover O_1 is greater than in (B,A).

Asymmetric Overload (Partition 3.c)

Referring back to Table 4.1 and Fig. 4.4, it is easy to verify the following.

Lemma 4.4.4. *In the case of asymmetric overload the sub-game perfect NE are*

$$\mathrm{NE} = \begin{cases} \{(A,A)\} & \text{if } (\rho_A,\rho_B) \in \textit{Region I} \\ \{(A,B)\} & \text{if } (\rho_A,\rho_B) \in \textit{Region II} \end{cases} \tag{4.11}$$

Finally, we note that the price of anarchy in the case of leader/follower deployment equals that of the simultaneous deployment. This can be easily seen by comparing the set of NE to the set of sub-game perfect NE.

4.5 Performance Evaluation

In this section, we validate the analytical model through simulating scenarios where two RSUs are either deployed simultaneously, or according to a leader-follower strategy. In the latter case, we consider the cases where RSUs are placed in different locations or are colocated. We designed the tests following the reference scenario outlined in Section 4.2, and implemented them in the ns-3 simulator. All vehicles travel at the constant speed of 20 m/s. After reaching the opposite end of the road, each vehicle is removed from the simulation. We make the conservative assumption that vehicles OBUs communicate with RSUs using the IEEE 802.11 MAC protocol at the basic rate of 6 Mb/s, regardless of the distance from the RSU, and that the coverage area of an RSU is 200 m. We use the default 802.11a implementation in ns3 based on Orthogonal Frequency Division Multiplexing (OFDM). At 6 Mb/s rate, Binary Phase Shift Keying (BPSK) is employed using 1/2 convolutional coding (100% redundancy). The physical layer model is as defined in YANS (Yet Another Network simulator [44] which has a channel model with a delay equal to the speed of light, and loss based on a log-distance model (46.6777 dB at 1m). The medium access layer in the implementation uses the Distributed Coordination Function (DCF) in 802.11a. The mobility model is chosen to be the constant velocity model within ns3.

Upon transiting under the coverage of RSUs, an OBU first listens for their beacon (transmitted every second), then tries to associate with one of them (picked randomly if more than one beacon is received). Finally, if successful, it starts uploading its content to the selected RSU by using MAC frames that can carry 1 kB of application data. If the transfer completes before the vehicles leaves the RSU coverage, the transfer is marked as successful. Otherwise, it counts as a failure, and the OBU will try to repeat the procedure upon coming under the coverage of another RSU (if any).

It is worth pointing out that, although the complexity (and realism) of the simulation scenario is at odds with the simplifications introduced by our game-theoretic approach, the purpose of our evaluation is to show that analysis still qualitatively captures the main trends observed in simulation.

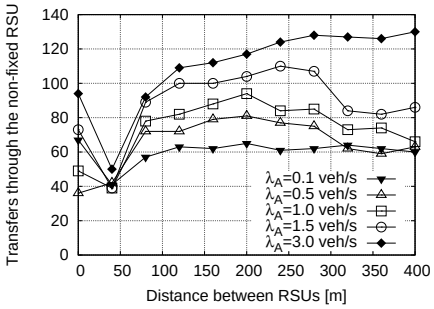


Figure 4.5. Successful content transmissions as a function of the distance between RSUs

In the first set of results, we investigate the revenues of two operators who simultaneously deploy their RSUs, by plotting the successful content transfers as a function of the distance between each operators' RSU. We assume that RSUs use the same frequency channel, which results in interference if the coverage areas overlap. The content size is fixed at 100 kB for each vehicle (translating into 100 MAC-layer frames that one vehicle must upload to an RSU for a transfer to be successful).

In Fig. 4.5, we plot the successful content transmissions by the non-fixed RSU as a function of its distance from the fixed one. Different left-to-right traffic intensities are considered, while the right-to-left traffic is kept constant. High and low λ_A, corresponding to the high and low traffic regimes described in Section 4.2 confirm the choice of, respectively, maximum RSU separation and colocation as the best strategies. In the "traffic stealing" case, corresponding to $0.5 \leq \lambda_A \leq 1.5$, choosing a location where RSU coverages overlaps plays into the hands of the non-fixed RSU, which can collect left-to-right spillover traffic in addition to serving all traffic in the opposite direction. Note that the dip at the 40-m RSU distance is caused by failed transfers to one RSU (the fixed one) that could not be completed on the non-fixed RSU, either. Indeed, left-to-right traffic associated to the fixed RSU starts a new upload to the non-fixed one only when fixed-RSU beacons are no longer received (200 m away from it). However, after an additional 40 m, the non-fixed RSU is out of range too. Such behavior is peculiar of the simulation scenario, and, thus, is not captured by the theoretical analysis. Similarly, the random ordering of successful transfer count at different traffi intensities in the colocation case (RSU distance = 0) is a result of vehicles randomly selecting the RSU to which they associate, as stated above.

In the second set of results, we quantify the revenues of each operator in case one of them is the market leader and the other does not have the first-move advantage. We consider two coverage scenarios: *colocated* and *disjoint*. The colocated setup refers to both RSUs occupying the same candidate site (either A or B) and transmitting on the same channel, while the disjoint (non-colocated) deployment is the one depicted in Fig. 4.1, where the candidate sites are 600 m apart. For each plot, we show several curves: "loc. A, alone" (resp. "loc. B, alone") representing the performance recorded by the RSU in location A (resp. B) in the disjoint deployment; "colocated" representing the performance of one RSU in the colocated deployment; "tot. disjoint" (resp. "tot. colocated") compounding the performance of the two RSUs in the disjoint (resp. colocated) case.

We initially study the case where a fixed ratio exists between left-to-right and right-to-left vehicle arrival rates (i.e., $\lambda_a/\lambda_b = 10$). In Fig. 4.6 we plot the number of successfully transmitted contents (whose size is fixed at 500 kB for each vehicle).

It is interesting to observe that, with these settings, whoever occupies location A first has the upper hand at low to medium traffic intensities. The higher number of vehicles flowing from left to right is turned into higher revenues for the operator in location A in the disjoint case. A new operator can hope to match the incumbent's revenues by

colocating if the RSUs are underloaded (λ_A <0.6 veh/s). These results match the model predictions for the *Colocation Underload* case.

If the vehicle arrival rate increases, colocation is not a good choice for the newcomer, who fares better on its own, i.e., non-colocation becomes the NE as predicted in the *Overload* case. Interestingly, the disjoint solution provides an advantage for the incumbent who selected location A only up to (λ_A <1.5 veh/s): at higher rates, its RSU cannot serve all vehicles and the spillover benefits location B, which again matches the performance of its competitor.

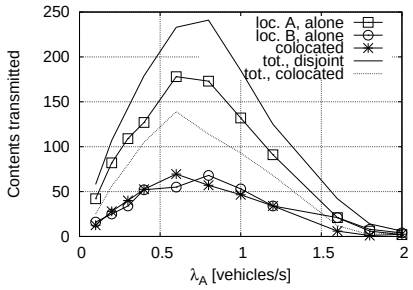


Figure 4.6. Constant arrival imbalance: successful transmissions as a function of left–to-right vehicle flow intensity

We next consider the case where the imbalance between the flows in opposite directions varies (i.e., λ_b is kept constant at 0.05 veh/s), while results in Fig. 4.7 are plotted for values of λ_a ranging between 0.1 and 2 veh/s. The content size is again fixed at 500 kB for each vehicle.

It can be seen that, as the rate of the left-to-right flow increases with respect to the opposite direction, location A becomes preferable. The newcomer's best choice is location B while λ_a is below 0.5 veh/s (which matches the *Colocation Overload (3.b)* case). However, for higher rates, colocation becomes preferable for the newcomer, as it guarantees more successful transmissions, as predicted by the *Colocation Overload (3.a)* case. We also remark that, as expected, this NE is not socially-optimal, as can be seen by the much higher combined revenues in the disjoint case.

Our final set of tests addresses the case of variable content size, which results in variable load offered to the RSUs. The arrival rates are fixed at λ_a =0.5 and λ_b =0.05 veh/s. Recall that if an OBU does not complete the transfer before leaving the RSU coverage, it will try afresh at the next RSU. This is shown in Fig. 4.8, where a sudden surge in offered traffic at location B can be detected for content sizes in excess of 300 kB.

A final look at Fig. 4.9 reveals that, for the chosen arrival rates, the content size increase does not affect the equilibria, but merely closes the gap between revenues achievable at disjoint locations.

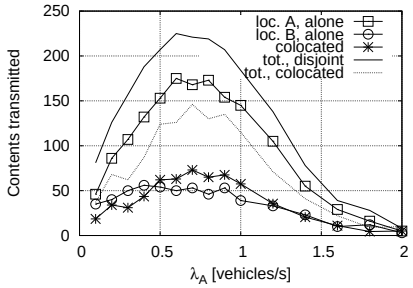


Figure 4.7. Variable arrival imbalance: successful transmissions as a function of left–to-right vehicle flow intensity

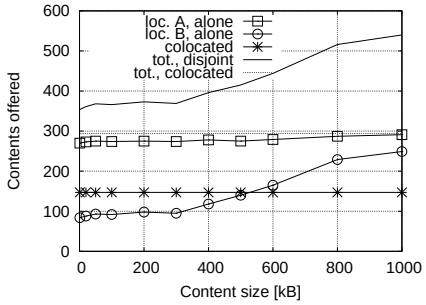


Figure 4.8. Variable content size: number of offered contents as a function of the content size

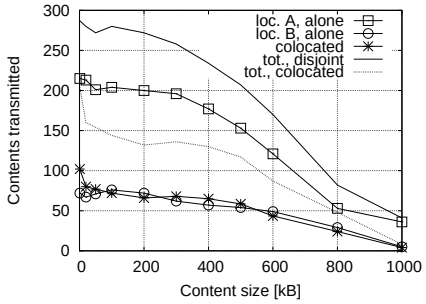


Figure 4.9. Variable content size: number of successfully transmitted contents as a function of the content size

4.6 Conclusions

In this chapter we looked at the problem of infrastructure deployment in VANETs through the lenses of game theory. We considered both simultaneous as well as leader-follower deployment, and quantified the inefficiency of equilibrium deployments compared to the social optimum. We then verified through simulations that, notwithstanding the necessary simplifications, our model correctly predicted the reachable equilibria as a function of traffic intensity and content size.

Chapter 5

Content sharing through the match-making paradigm

5.1 Introduction

Once the network is deployed, we need to devise an efficient way to manage it, i.e., to make users able to quickly and effectively fetch the content they need.

The network dynamics in a wireless environment, however, are different from their wireline counterpart. Node churning, for example, is a common hurdle in peer-to-peer systems for wireline networks: mobility and variable channel conditions in wireless networks only exacerbate it. Thus, it is of paramount importance that the content carried by mobile users is easily, promptly "discoverable" and that its carriers are reliable when it comes to providing the content to others.

To this end, an efficient content discovery paradigm for mobile networks is needed. One possible candidate is the publish/subscribe (pub/sub) paradigm, which provides for an asynchronous content exchange between publishers (providers) and subscribers (consumers). Content attributes are specified by publishers and, through filtering techniques, subscribers are delivered content whose attributes match constraints defined by them. However, in pub/sub systems the implication is usually that content is delivered to subscribers as soon as it becomes available through one or more publishers. Given the fleeting connectivity they experience, this behavior may quickly lead to bandwidth waste and low hit probability. An alternative is represented by quorum-based replication schemes, where content update and request operations are carried out in interacting subsets of nodes, called read quorum and write quorum. Again, although specifically designed for distributed systems, quorum schemes are hardly a good choice in mobile networks, mainly due to the overhead they generate and the complexity in controlling the topology.

In this work we take a different approach. We present a content discovery solution,

called Figaro, where mobile users are supported by an infrastructure – a scenario that finds wide application in the real world. In Figaro, mobile users, named Agents, request content items of their interest and, in their turn, make content items available to others. To ease the information sharing, users advertise, i.e., inform infrastructure nodes, named Brokers, about which content they are willing to provide, and Brokers assist requesting Agents in the content discovery process. To distinguish Figaro from standard pub/sub systems, we refer to its paradigm as *match-making*, highlighting its capability to let demand and offer meet *when the need arises*, while arbitrating the information flow between providers and consumers to account for the specific characteristics of mobile networks. Also notice that, in contrast with the pub/sub paradigm, the Agents selected as providers do not have the possibility to (legitimately) refuse to provide a content. In Figaro, the underlying assumption is that the Broker "knows better", i.e., it has a most reliable knowledge of whether an Agent has to provide a service.

One of the most important performance metrics we consider in our system is the query success probability, that is, the probability that a content query is matched with an Agent that owns the desired content and is willing to provide it. To ensure high success probability for content queries as well as a fair treatment to users, we act as follows:

- We associate to each Agent a *credit balance*, which increases when the Agent provides a requested content and decreases when it consumes a content. The same concept has been previously exploited to favour traffic routing in ad hoc networks [45]. We revise this approach and apply it to content discovery, showing that it can make content provisioning a rational choice for self-interested Agents, hence it can discourage rational Agents from acting as free riders. Furthermore, by letting the Agents' balance increase/decrease depending on the size and popularity level of the provided/requested content, we can guarantee fairness in spite of the different characteristics of the information advertised or requested by the Agents.

- We define a feedback mechanism that allows Brokers to identify and ban those Agents that do not provide the content they advertise. They can be either rational Agents acting as free riders or malicious users that aim at disrupting the system. We will refer to the latter as *disruptors*, since, regardless of whether they generate content requests or not, their main goal is to disrupt the success probability of queries issued by others. The feedback mechanism is designed so that Brokers can detect and discard those negative feedbacks that are likely to be part of a bad-mouthing attack, and, as discussed later in the chapter, it is only marginally vulnerable to other kinds of attacks.

- To guarantee a fair treatment to Agents providing content items with different characteristics, as well as to evenly distribute the load of content provisioning, we exploit Agents' caching capabilities. We formulate caching as an optimization

problem that aims at maximizing the system fairness and efficiency, and design a heuristic that closely approximates the optimal solution while accounting for the system dynamics.

The rest of the chapter is organized as follows. Sec. 5.2 reviews previous work, highlighting the novelty of our contribution. Sec. 5.3 describes our content discovery scheme Sec. 5.4 introduces the credit, feedback and banning mechanisms, whose effectiveness is analysed in Sec. 5.5 through game theory. The resilience of Figaro to different attacks is discussed in Sec. 5.6, and confirmed by the results we obtained in Sec. 5.7. Sec. 5.8 introduces our caching strategy, while Sec. 5.9 shows its effectiveness in providing high query success probability and fairness, and it compares the performance of Figaro with some existing solutions. Finally, Sec. 5.10 concludes the chapter.

5.2 Related Work

Our match-making paradigm draws from the pub/sub paradigm, which has been extensively studied in the literature. However, most works focus on wired scenarios, or on wireless ad hoc networks without any infrastructure. The opportunities offered by the presence of an infrastructure in a wireless environment are investigated in [46], which, however, does not address fairness, cooperation, or caching.

Associating network nodes with a balance is an idea that has been often exploited to enforce cooperation among self-interested nodes in wireless ad hoc networks, either for traffic routing [45, 47] or for channel access [48]. Note, however, that the seminal work in [45] requires the nodes to be equipped with a tamper-resistant hardware (i.e., a security module manufactured by a limited number of trusted manufacturers), in order to prevent attacks. The study in [47], instead, does not deal with attackers at all. More recent works, e.g., [49, 50], still rely on the assumption that a security module is available, and propose a distributed incentive protocol for multi-hop routing in mobile networks. We point out that in Figaro nodes are not required to embed any tamper-resistant device; indeed, through a balance- and feedback-based mechanism, the scheme itself ensures resilience to both free riders and attackers.

With regard to feedback-based schemes, of particular relevance is the pioneering work in [51], which introduces a reputation mechanism to enforce cooperation among rational nodes of a mobile ad hoc network. Many later studies have focused on cooperative routing in ad hoc networks [52] and in overlay networks [53]. Note that the proposed solutions refer to a different type of cooperation with respect to Figaro, i.e., message forwarding instead of content transfer. Consequently, the attack they consider is packet dropping, while in Figaro we address the problem of nodes that do not provide an advertised content when requested by the Broker. Furthermore, in Figaro there is no need for sophisticate misbehavior detection and identification schemes of misbehaving

nodes, as the Agents know exactly when they are victim of either a free rider or a disruptor, and can notify the Broker of the identity of the attacker (i.e., the Agent who did not to provide them with a content).

At the application layer, solutions for content provisioning have been presented in [54], where a reputation-based scheme is used to reduce the load over a 3G network. Monetary penalties and incentives are given to non-cooperative and caching nodes, respectively, while the choice of the content to cache is left to the Agents and modeled as a market sharing game. We point out that, unlike previous work, our approach is simple, lightweight and it does not assume that Agents are associated to a billing account: in Figaro incentives and penalties are circumscribed to Figaro itself and, since Agents do not directly choose which peer they will retrieve content from, Figaro has high resiliency to reputation attacks.

As for caching, again most schemes designed for wireless networks, e.g., [55, 56], focus on distributed, infrastructure-less scenarios. As a result, they imply a complexity level that is exceedingly high for Figaro, whose aim is to leverage the presence of Brokers and their centrality in the system architecture to simplify the network management.

Finally, we mention BubbleStorm [57], a well-known scheme for content replication and provisioning, which is based on a probabilistic exhaustive search paradigm in wired overlay networks. Unlike Figaro, BubbleStorm assumes the nodes to be always willing to store a copy of the content. As soon as it is generated, BubbleStorm propagates the content on a random graph defined on the overlay network. Content queries are propagated following the same strategy, and they succeed if at least one copy of the query reaches a node that stores a copy of the content. In Sec. 5.9.2, we will use BubbleStorm as a benchmark for the performance of Figaro.

An early version of this work, sketching the match-making paradigm for content discovery, can be found in [58].

5.3 The Figaro System

We envision Figaro as an overlay network that operates according to the match-making paradigm. At a *logical* level, Figaro features two main types of nodes: Agents and Brokers. Agents store, advertise and consume content items, while it is the Brokers' task to let demand and offer meet. Agents store either self-produced content (e.g., a set of pictures one wants to share) or content they have a specific interest to spread (e.g., information on a film festival one is interested in). Content items are assumed to be relatively static, i.e., they are updated at intervals that are much longer than an interaction between Agent and Broker, e.g., once a day.

At a *physical* level, Agents are mobile (possibly hand-held) devices, while Brokers are middle-end devices, integrated in an infrastructure and interconnected via a reliable backbone. From the viewpoint of network connectivity, we consider each Broker to be

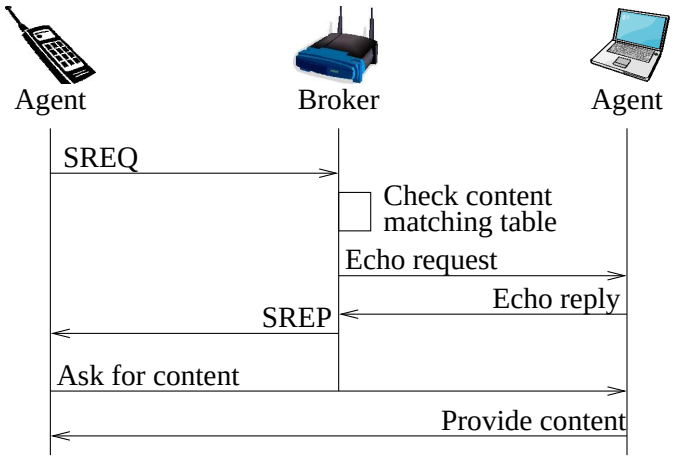


Figure 5.1. Basic message exchange between Agents and Broker.

colocated with a router and to be associated to an IP subnet. One or more IEEE 802.11 Access Points (APs) are attached to the same router interface and provide wireless connectivity to mobile nodes within the area. Thus, the mobile nodes are hosts of the router subnet to which the Broker is associated.

An Agent becomes part of the Figaro system when the mobile device embedding it associates to an AP and discovers the Broker that controls it. The Agent can choose to register with this Broker, advertising the content it is willing to share with others. The Broker maintains a content-based matching table, where it stores the following information for each Agent: 1) a unique Agent identifier; 2) its IP address; 3) the MAC addresses of any interface (e.g., 802.11, Bluetooth) the mobile node carries; 4) the content it makes available to others. The set of Agents registered to the same Broker is called *Colony* and it corresponds to the hosts of the subnet associated to the Broker (i.e., at the IP layer, all Agents of a Colony have the same subnet ID as the Broker).

A registered Agent can ask the Broker to identify another Agent carrying the content it needs, through a Service REQuest (SREQ) message. The Broker queries its own content-based matching table to identify a candidate Agent that can provide such content. As detailed in Sec. 5.4, these Agents are selected as candidate according to Colony-wise policies, aimed at pursuing specific objectives (e.g., high success probability and even load distribution on Agents). Next, the Broker checks that the candidate provider Agent is still reachable by a `ping` at the IP address with the Colony subnet ID, otherwise it selects another candidate provider from the matching table[1]. To avoid unpredictable iterations, the selection does not account for lower-layer metrics, such as the SNR on the links between Agents and AP, which can only be established upon

[1]Deregistration of an Agent is autonomously enforced by a Broker after the Agent is found unresponsive to a certain number of consecutive attempts at pinging it.

checking the provider Agent's reachability.

If a candidate provider Agent is found, the Broker returns to the querying Agent a Service REPly (SREP) message carrying the IP address and the MAC address(es) of the candidate provider Agent. A transport-layer connection between the two Agents for the purpose of content transfer is subsequently established. Such a connection runs within the coverage area of the APs connected to the router interface.

If, instead, no candidate provider is found or if none of them replies to the `ping`, the search is relayed to a higher hierarchical level. To this end, we introduce an architectural entity, called Proxy, which is connected to all Brokers via the backbone. When a Broker receives a request for a content that is unavailable in its Colony, it forwards the request to the Proxy, which in turn queries the other Brokers. Also, when an Agent moves to a different Colony, the new Broker informs the previous one of the Agent's migration. For scalability reasons, a hierarchy of Proxies can be deployed, although we leave it out of the scope of this chapter. The request is successful if the content is found in any of the Colonies composing Figaro. In this case, the connection between Agents runs through the routers colocated with the Brokers that control the Agents.

Agents report to their Broker the outcome of successful and unsuccessful content transfers with Agents identified as candidate providers. The outcome is notified through a *feedback* message, which the requesting Agent sends to the Broker after the content transfer.

5.4 Matching Demands and Offers

We consider a mobile system where Agents are rational and follow the same behavior in terms of querying activity. Let I be the number of content items that exist in a Figaro system composed by C Colonies; the items may differ by size and popularity. Let Λ be the per-Agent query generation rate. Upon a query generation, an Agent selects the item to ask for according to its popularity level, i.e., with probability $\pi(j,t)$, $1 \leq j \leq I$ ($\sum_{j=1}^{I} \pi(j,t) = 1$). Consequently, at time t within Colony k, each content j is requested with rate $\lambda_k(j,t)$, which is equal to $\Lambda\pi(j,t)$ multiplied by the number of Agents in Colony k at time t. We also denote by $P_k(j,t)$ the number of Agents (either under or out of coverage) advertising item j at time t in Colony k.

We design our match-making system in order to achieve the following goals:

- high query success probability, in spite of the rational behavior of users and the different characteristics of the requested content (i.e., size and popularity level);
- fair treatment of the Agents, i.e., the amount of service they provide is comparable to the amount of service they obtain;

- resilience to Agents who do not provide the content they advertise (either free riders or disruptors).

To meet these objectives, we associate to each content j a Colony-wise value, denoted by $G_k(j,t)$, which is expressed in credits and may vary in time: an Agent that provides (receives) a content item earns (spends) an amount of credits equal to the content value. For each Agent i, we can therefore define a balance $b(i,t)$, expressed again in credits, which reflects the difference between the value of the content the Agent has provided and the value of the content it has obtained. Note that the exact, up-to-date value of each Agent's balance is only known by the Broker (although the Agents can compute their own rough estimate).

We define $G_k(j,t)$ so as to take into account both the different size and popularity level of the content items. More specifically, for each Colony k we introduce the *content burden* metric, $B_k(j,t)$, which is the ratio of the query rate associated to content j within the Colony to the number $P_k(j,t)$ of Agents providing it at time t, i.e.,

$$B_k(j,t) = \frac{\lambda_k(j,t)}{P_k(j,t)} \tag{5.1}$$

Note that, in defining $B_k(j,t)$, we consider the query rate for content j coming from other Colonies to be negligible. Also, the burden takes larger values for popular content (characterized by high values of $\lambda_k(j,t)$) and for rare content (for which $P_k(j,t)$ is low), while it is smaller for content with low popularity or that can be easily found in the Colony. By denoting with $s(j)$ the size of the file representing content j, we define $G_k(j,t) = (g + B_k(j,t))s(j)$, where $gs(j)$ represents the baseline value of the content. It follows that large-sized, highly-popular content items, as well as rare items, will all be highly valuable. Also, we associate a value of g credits to each feedback message that an Agent belonging to Colony k sends to the Broker at time t to notify the outcome of the transfer of the requested content.

For the sake of clarity, let us first consider the case where there are no malicious Agents, and assume that Agents start with a zero balance. The balance $b(i,t)$ of the generic Agent i, belonging to Colony k, is updated as described below.

- When Agent i requests a content j by sending an SREQ to the Broker and the Broker finds a candidate provider:
  (i) $b(i,t)$ is decreased by $G_k(j,t) + g$;
  (ii) if Agent i sends a feedback to the Broker, related to the transfer outcome of a requested content, $b(i,t)$ is increased by g;
  (iii) if Agent i is not satisfied by the transaction (e.g., the transfer fails to complete or does not occur at all), it sends a negative feedback; it is entitled to request the same content again (provided that the new query is made within a given time

window) without further decreasing its balance[2].

- When Agent i is selected as provider for a content query issued in Colony l ($l \neq k$):
  (i) $b(i,t)$ is increased by $G_l(j,t)$;
  (ii) if a negative feedback about the data transfer is received by the Broker, $b(i,t)$ is decreased by $G_l(j,t)$.

Note that feedback messages play a very important role in Figaro's credit scheme and are therefore awarded additional credits: by doing so, rational Agents will always provide a feedback if they can. Also, Agents have no incentive to provide a falsely negative feedback, as this would not restore their balance, but only give them the opportunity to request the same content again – which would be useless if the content has already been successfully received. In case no feedback is received for a content transfer, the transfer is assumed to be successful (i.e., the candidate provider is awarded $G_k(j,t)$ credits). The rationale is the following: since requesting Agents always have an interest in sending a feedback, connectivity problems likely prevented the Broker from receiving a feedback. If this were the case, the same connectivity problems would be the cause of the transfer failure (if any), and taking actions against the candidate provider would be unnecessary.

Given the above credit scheme, the Broker can exploit the value of the Agents' balance to ensure Agents' cooperation in providing content. In particular, the Broker can determine whether a requesting Agent is entitled to receive further service and which Agent should be selected as candidate provider, according to the following rules:

- upon receiving an SREQ from Agent i, the Broker discards the SREQ if $b(i,t) < T_r$, where T_r is a negative threshold value (i.e., the requesting Agent has too low a balance to request a content);

- otherwise, a candidate provider is selected and the Broker appoints the Agent that has the lowest balance among the Agents advertising the requested content.

The credit system described above makes cooperation, i.e., providing a content when requested by the Broker, necessary for the Agents in order to be able to obtain the content they need later on. The higher the T_r, the higher the amount of cooperation required. When all Agents are rational, game-theoretic methods can be used [47] to assess the value for T_r that yields optimal performance. However, in Figaro we also take into account the presence of malicious Agents, whose only purpose is to disrupt the system performance. To counter them, we introduce a banning mechanism, which changes the nature of the problem and, as a positive side-effect, also represents a further incentive for rational Agents to cooperate.

[2] Subsequent feedbacks related to the same content do not bring any further increase in the balance of the issuing Agent.

5.4.1 The banning mechanism

Figaro uses banning to keep malicious Agents out of the Colony and, thus, impair their actions. Every time the Broker receives a negative feedback related to an Agent (and it deems it credible, as described below), that Agent is banned for a certain period of time. While banned, the SREQs transmitted by the Agent are dropped by the Broker and the Agent cannot be selected as a candidate provider. We stress that when a banned Agent issues a SREQ, its balance is decreased anyway by the value of the requested content. Recall that Agents are not aware of their balance. To prevent an Agent from foreseeing the ban periods and avoiding to request content items while being banned, ban periods start after a random time since banning is triggered.

The ban duration grows exponentially: on the n-th time that an Agent is banned, the duration is computed as $T_b(n) = T_0 a^{n-1}$, with $a \geq 1$,$n \geq 1$. T_0 is set small enough so as not to excessively penalize those Agents that occasionally fail to provide a content, and a large enough so as to rapidly and effectively exclude malicious Agents from the Colony. The actual choice of T_0 and a depends on the application and, as shown later in the chapter, on the system status; also, the counter recording the number of bans for each Agent can be periodically reset.

We point out that the ban mechanism not only allows Brokers to counteract malicious Agents but it also serves as further incentive for rational Agents to provide the requested content when selected as candidate providers. This makes our study significantly different from the one in [47]. Furthermore, Figaro is immune to the adverse impact that mobility may have on the effectiveness of reputation and ban schemes: banned Agents may try to move to a different Colony to nullify their banning, or newly arrived Agents may be unable to figure out the trustworthiness of their neighbors. Indeed, in Figaro (i) Brokers can exploit the backbone to exchange information about banned Agents or about the balance of Agents that move from one Colony to another, and (ii) newcomers (like all other Agents) rely on Brokers for the selection of the candidate provider.

5.4.2 Feedback credibility

As is evident from the description above, feedbacks are of primary importance for both balance update and ban. Thus, in Figaro bad-mouthing attacks, in which attackers assign falsely negative feedbacks to the Agents that provide them with a content, would cause a serious malfunctioning. To ensure robustness to bad-mouthing attacks, each Broker implements a simple, yet effective, credibility filter, based on the notion of *negative feedback ratio*.

We begin by introducing some definitions. Given Agents u and v, the negative feedback ratio $\nu_I(u,v)$ is the fraction of negative feedbacks issued by Agent u on Agent v's behavior. We then define the following Colony-wise average values: $\bar{\nu}_I(u) =$

$\frac{1}{N_u}\sum_v \nu_I(u,v)$, i.e., the ratio of negative feedbacks issued by u averaged over the number (N_u) of nodes that have served as candidate providers for u and belonging to the same Colony as u; $\bar{\nu}_R(v) = \frac{1}{N_v}\sum_u \nu_R(u,v)$, i.e., the ratio of negative feedbacks received by v averaged over all Agents belonging to the same Colony as v and for which v has acted as candidate provider.

Then, let us consider that the Broker receives from Agent u a feedback on the content provider v, and both u and v belong to the Broker's Colony. The Broker deems the feedback to be not credible if both the conditions below hold:

1. the ratio of negative feedbacks issued by Agent u, $\bar{\nu}_I(u)$, is higher than the average value computed over all Agents belonging to the same Colony as u (i.e., the issuer's view of the Colony is more negative than the average one);

2. the ratio of negative feedbacks given by u to v is higher than the average negative feedback ratio received by v: $\nu_I(u,v) > \bar{\nu}_R(v)$ (i.e., the issuer's view of the provider Agent v is more negative than the average one).

If, instead, the candidate provider v and the Agent u issuing the feedback belong to different colonies, typically $\nu_I(u,v)$ is not statistically meaningful due to a small number of occurrences. Hence, the Broker of Agent u only evaluates the first condition and notifies the outcome to the Broker of Agent v. Based on this condition only, the Broker of Agent v assesses the feedback credibility. Notice that, when a feedback is considered not credible, it does not trigger the banning of the candidate provider and no action is taken against its issuer. The rationale of the latter choice is that, once the bad-mouthing Agent is identified and made harmless, the Colony can still benefit from its presence, as long as it correctly provides its content when asked. Also, Agents found to issue unreliably negative feedbacks are not necessarily attackers; they may simply be Agents whose ability to receive content items is impaired by some external reason (e.g., connectivity issues): banning them would be unfair.

In the following, we highlight the ability of the presented credit scheme and candidate provider selection policy to ensure a high query success probability, and we discuss the robustness of Figaro to the possible attacks by malicious Agents.

5.5 Ensuring Cooperation in Figaro: A Game-theoretic Analysis

We now adopt a game-theoretic approach to show that our credit scheme, jointly with the banning mechanism, make cooperation (i.e., providing the requested content when selected by the Broker as a candidate provider) the best choice for a rational Agent. We therefore focus on rational Agents and assume that none of them is malicious.

We model the system dynamics as a game, where, when selected by the Broker as candidate provider, an Agent can play two possible moves: to provide or not to provide the content. We first compute the payoffs corresponding to these moves. Then, we derive the strategic form of the game and show, by iterated dominance, the condition under which there is a unique Nash equilibrium, in which all players cooperate. Finally, we show that such an equilibrium is Pareto-optimal and also attains the maximum efficiency.

5.5.1 Payoffs and game solution

We first assume homogeneous conditions, that is, independently of the considered Colony, all content items are represented by a file of the same size and have the same popularity level, and for each content there is an equal number of Agents storing the content. Hence, $G_k(j,t) = G(t), \forall k,j$. We will then extend the analysis to the inhomogeneous case where content items have different characteristics.

We consider a generic Agent i belonging to Colony k and assume that, at a generic time t, it is selected as a candidate provider. Let $V(i,t)$ be the utility that Agent i can expect to obtain, i.e., the amount of service it will be able to receive in the future (not considering the possibility to be subsequently selected as a candidate provider). In Figaro, this corresponds to having $V(i,t) = b(i,t)$, i.e., the Agent's current balance. We denote with $V^+(i,t)$ and $V^-(i,t)$ the new utilities of Agent i in case it chooses, respectively, to cooperate and not to cooperate at time t. Also, let $c(j)$ be the cost of providing content j. The cost $c(j)$ is assumed to directly reflect the size $s(j)$ of the file representing content j, e.g., $c(j) = Ks(j)$, where K is a constant positive value. However, due to the assumption of homogeneous content, $c(j) = c = Ks, \forall j$.

Now, if Agent i decides to cooperate, it pays a cost c for providing the content, and its utility $V^+(i,t)$ will change due to the increase of its balance by $G(t)$. Its payoff will be:

$$U_c(i,t) = -c + V^+(i,t) = -c + [G(t) + b(i,t)]\,. \tag{5.2}$$

Conversely, if Agent i decides not to cooperate, it will not pay any cost. However, its utility $V^-(i,t)$ will reflect the fact that not only will its balance $b(i,t)$ not increase, but the Agent will also be banned for a time interval, whose duration depends on the number of bans already received. Let us assume that the Agent has already been banned $(n-1)$ times; considering that, during the ban period, it will issue an average of $\Lambda T_b(n)$ SREQ messages, and a decrement of its balance will correspond to each of them, its payoff becomes

$$U_{nc}(i,t) = V^-(i,t) = \max\Big\{\,[b(i,t) - \Lambda T_b(n) G(t)]\,, 0\Big\}. \tag{5.3}$$

Note that, obviously, an Agent cannot retrieve less than 0 items thus $V^-(i,t) \geq 0$.

Observe that this is a rather peculiar game, as each player's payoff solely depends on its own move, and not on the opponent's one. As far as the equilibrium is concerned, the game can be easily solved by iterated dominance [59]:

- if $U_c(i,t) > U_{nc}(i,t)\ \forall i,t$, (Cooperate, Cooperate) is the sole Nash equilibrium;
- if $U_c(i,t) < U_{nc}(i,t)\ \forall i,t$, (Not cooperate, Not cooperate) is the sole Nash equilibrium;
- if $U_c(i,t) = U_{nc}(i,t)\ \forall i,t$, there is no unique Nash equilibrium.

In order for (Cooperate, Cooperate) to be a Nash equilibrium, we need $U_c(i,t) > U_{nc}(i,t)\ \forall i,t$. From (5.2) and (5.3), we obtain the following condition:

$$T_b(n) > \frac{c - G(t)}{\Lambda G(t)} \quad \forall t,n. \tag{5.4}$$

In other words, if the condition in (5.4) is met, the game will reach an equilibrium in which all rational Agents always cooperate, i.e., they provide the content the Broker asks of them. Condition (5.4) must hold for any time instant t and for every n; thus, considering the expressions of c and $G(t)$, a possible choice for T_0 is given by: $T_0 > (K - g)/\Lambda g$. Such an expression can be read as follows: the higher the request rate Λ, the more severe the penalty that banned Agents receive during the ban period and, thus, the smaller the value of T_0 needed to make cooperation a convenient strategy. Clearly, if $K \leq g$, banning is not necessary to make cooperation a rational choice for the Agents.

5.5.2 Optimality and efficiency of the operational equilibrium

When the condition in (5.4) holds, the strategy profile (Cooperate, Cooperate) is a Nash equilibrium and, since $U_c(i,t) > U_{nc}(i,t)$ for every Agent i, it is also Pareto optimal, i.e., cooperation is the best strategy that an Agent can follow without making someone else worse off.

In order to assess the efficiency of this equilibrium, we determine the price of anarchy (PoA), which is defined as the ratio of the payoffs obtained by the players when the Nash equilibrium holds, to the payoffs obtained by the players if a globally optimal solution is enforced [59]. In the latter case (in which Agents are not free to chose whether they want to cooperate or not), an Agent that is selected as a candidate provider again pays the cost c while its balance is increased by $G(t)$. Thus, $U_{opt}(i,t) = -c + [G(t) + b(i,t)] = U_c(i,t)$, i.e., PoA $= 1$. In other words, not only is the equilibrium (reached when all Agents cooperate) fair and Pareto-optimal, but it yields the very same efficient behavior as the globally optimal solution. Intuitively, this is not surprising, since cooperation means that Agents follow the suggestions of the Broker, which is in an excellent position to determine the optimal strategy.

5.5.3 The inhomogeneous case

We generalize the previous analysis to the case where items have different size, popularity level and availability, and popularity and availability may depend on the Colony. Thus, we now consider a cost $c(j) = Ks(j)$ and a content value $G_k(j,t)$ ($1 \leq k \leq C$, $1 \leq j \leq I$).

In this setting, the payoffs for a cooperating and not cooperating Agent i, which belongs to a generic Colony k and is requested to provide content j at time t, are given by:

$$U_c(k,j,i,t) = -c(j) + V^+(i,t) = -c(j) + G_k(j,t) + b(i,t) \tag{5.5}$$

$$U_{nc}(k,j,i,t) = V^-(i,t) = \max\left\{\left[b(i,t) - \Lambda T_b(n)\sum_{h=1}^{I}\pi(h,t)G_k(h,t)\right],0\right\} \tag{5.6}$$

where we assumed that i has already received $n-1$ bans. In order for the Agents to cooperate, the condition $U_c(k,j,i,t) > U_{nc}(k,j,i,t)$ has to hold for any k,j,i and time instant t, i.e.,

$$T_b(n) > \frac{\max_j [c(j) - G_k(j,t)]}{\Lambda \sum_{h=1}^{I}\pi(h,t)G_k(h,t)} \quad \forall t,n. \tag{5.7}$$

Then, considering the expressions of $c(j)$ and $G_k(j,t)$, a possible choice for T_0, so that the condition in (5.7) is always satisfied, is given by:

$$T_0 > \frac{(K-g)\max_j s(j)}{\Lambda g \min_j s(j)}. \tag{5.8}$$

5.6 Resilience to Attacks

Malicious Agents may try to break Figaro's balance mechanism by performing several types of attacks. In particular, they may behave as disruptors with the sole purpose of degrading the system performance. Figaro counteracts this behavior through the ban mechanism, which leaves out of the system an Agent for a given period of time, as soon as it receives a credible negative feedback. The effectiveness of banning is shown in Sec. 5.7.2, through ns-3 simulation.

Below, instead, we discuss the resilience of Figaro to the typical attacks that may be launched (independently or in a collusive manner) in online trading communities or, more in general, in reputation-based systems.

Ballot stuffing: a group of Agents collude to give each other positive feedbacks, in order to get an incorrectly high balance. In Figaro, Agents cannot freely choose whom they ask for the content they need, since the selection is performance by the Broker.

Therefore, while it is possible to indiscriminately give a colluder a positive feedback, a large number of colluders is needed to make this attack viable (i.e., by increasing the likelihood that one of my colluders is selected as provider). Furthermore, an Agent has no practical way to artificially increase the number of feedbacks it is entitled to issue regarding its own colluders, since the number of requests it can issue is limited by T_r.

Bad-mouthing: a group of Agents collude to give negative feedbacks to others, so as to incorrectly lower their balance, and having them repeatedly banned. Again, the effectiveness of this attack is dampened by the Broker likely choosing a different provider for every request.

Negative/positive discrimination: an Agent provides the requested content only to a selection of other Agents, neglecting those it "does not like". This behavior will draw bans upon the Agent and is hardly effective in the long run.

Sybil attack: an Agent uses a large number of pseudonyms, thus gaining a disproportionately large influence on its own reputation scores, as well as the other peers'. However, Agents are identified via their IEEE 802.11 MAC address. An attacker could modify its MAC address to assume a new identity, but in this way its former identity would become unreachable and would be automatically de-registered from the colony, i.e., an Agent cannot have multiple *contemporary* identities, unless relatively complex hardware and software are used.

Whitewashing: an Agent misbehaves until it is banned and then assumes a new, clean identity. Whitewashing is normally impervious to detection attempts. The only protection that Figaro can deploy is by not letting Agents know they are banned (hence not letting the attacker know explicitly when it is time to assume a new identity). Still, Figaro is vulnerable to whitewashing by knowledgeable attackers who are aware that a negative feedback will get them banned: they can stay in the system until they are first requested to provide a content, ignore it and then assume a clean identity. Thus, in the case of applications for which resilience to whitewashing (and Sybil) attacks is highly critical, Agents may be required to perform a una-tantum, web-based registration (possibly with a CAPTCHA [60]) before they can register to a colony. Furthermore, Agents could be required to have a private key and sign the messages they send.

5.7 Performance Evaluation

We provide an evaluation of Figaro's features described so far by using ns-3 simulations: firstly, by looking at its resilience against several types of attacks, and then by establishing to which extent it can actually enforce cooperation and fairness among the Agents.

5.7.1 Network scenario

Our investigation focuses on a pedestrian scenario, where four points of interest (POI) are placed at the corners of a 1000×850 m^2 rectangular area. In correspondence to each POI there is an 802.11 AP, integrating a Broker, all connected through a backbone. Agents, whose number is fixed to 100 unless otherwise specified, are equipped with an 802.11 interface and roam among the APs. Their movement follows the Random Trip mobility model, with an average pause time of 100 s and an average speed of 1.8 m/s. The Two-ray Ground model is used to represent the channel propagation conditions, and the transmission data rate between Agents and APs is controlled through the AARF technique [29], so as to adapt to the perceived channel conditions.

The content items are divided into four classes with different popularity and size. We consider two possible levels of content popularity, as well as two possible content sizes. Both the content sizes and the popularity levels differ by a factor 2. Specifically, class 1 items have size of 200 kB and popularity level equal to $1/3$, class 2 items have size of 100 kB and popularity level $1/3$, class 3 items have size of 200 kB and popularity level $1/6$, and class 4 items have size of 100 kB and popularity level $1/6$. For the sake of clarity while presenting the results, we consider $I = 4$, i.e., one content per class, and assume that each Agent advertises exactly one content (chosen with equal probability among the possible four), so as to associate each Agent to the class of content it provides. We stress, however, that simulations with a larger number of content items yielded qualitatively similar results.

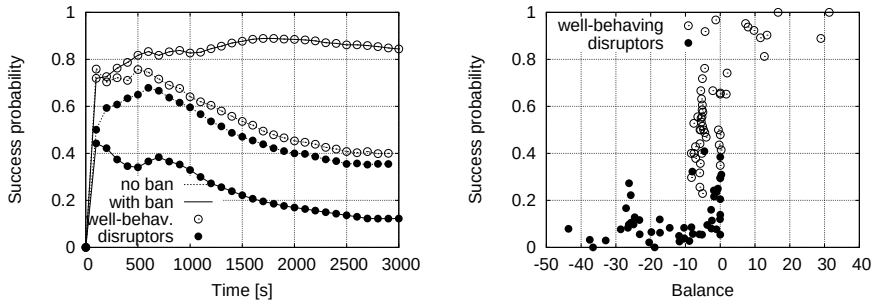
An Agent "becomes interested" in a content according to a Poisson process with rate equal to $\Lambda = 0.02$ req/s. The requested content j is chosen, among those not stored by the Agent, with probability proportional to the content popularity level $\pi(j,t)$. The Agent then issues an SREQ message for that content, which is periodically refreshed until an SREP is returned, or until a timeout (set to 30 s) expires. If no reply ensues before the timeout, the Agent considers the query as failed. Instead, if a positive reply is received from the Broker, the requesting Agent asks the providing Agent for the content. Data is exchanged through a well-known UDP port. Agents are not required to implement any routing protocol, as the content is either available in the same Colony (i.e., subnet) or via the backbone.

As for the other parameters, we set: $T_0 = 60$ s, $a = 3$, $K = 1$, $g = 0.5$. Finally, all plots shown in the following have been obtained by averaging 10 simulation runs.

5.7.2 Counteracting disruptors, bad-mouthers and liars

When our credit-based scheme is implemented, we have proved that all rational users have interest in cooperating, hence they will provide the content when they are asked for. Here, we are therefore interested in evaluating to which extent Figaro can (1) protect well-behaving Agents from disruptors; (2) detect and discard falsely negative feedbacks,

i.e., the ones issued by Agents taking part in a bad-mouthing attack; (3) make it disadvantageous for Agents to lie about the content they share in the Colony. Note that the plots presented here do not show the class of the content the Agents provide, as this is not significant for the aspects being taken into account.



(a) (b)

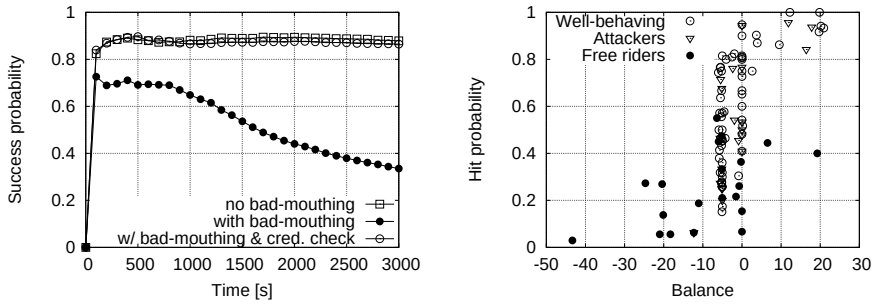
Figure 5.2. Resilience to disruptors: (a) time evolution of the success probability for well-behaving Agents and disruptors, with/without banning; (b) success probability vs. balance for well-behaving Agents and disruptors, with banning enabled.

We start by considering a scenario in which 50% Agents are willing to cooperate while the rest are disruptors. In our simulations, we consider that disruptors also issue content requests and that their behavior is unaffected by the credit and ban mechanisms, and we set $T_r = -5$ (the impact of T_r will be evaluated later).

First, to show the effectiveness of our banning mechanism, Fig. 5.2(a) presents the time evolution of the query success probability of well-behaving rational Agents and of disruptors, with and without banning. In absence of banning, well-behaving Agents and disruptors experience about the same success probability: the success probability decreases over time till a saturation value (namely, about 0.4), which is determined by the presence of a large percentage of disruptors[3]. Conversely, with banning, the Broker can tell apart well-behaving nodes and disruptors in a very reliable manner, and the gap in performance between the two types of Agents widens. Indeed, with the passing of time, more and more disruptors are discovered and banned, hence made harmless. Consequently, over time the probability that a disruptor is elected as candidate provider decreases while the success probability of well-behaving Agents grows. However, the success probability of well-behaving Agents does not reach 1 and, similarly, the success probability of disruptors does not drop to 0. This is due to the following reasons: (i) the ban period of disruptors is limited, thus, at any time instant there may be still disruptors active in the Colony (although they will be detected and banned again); (ii) as studied in the next section, the balance of the Agents providing lower-value items (classes 3-4)

[3]The more the disruptors, the fewer the Agents providing an advertized content, hence the lower the success probability.

may drop below the threshold T_r (i.e., they cannot gain as many credits as they would need to obtain the desired content).



(a) (b)

Figure 5.3. Resilience to bad-mouthing: (a) time evolution of the success probability of well-behaving Agents, with/without bad-mouthing. The cases with and without feedback credibility check are shown; (b) query success probability vs. balance, for well-behaving, bad-mouthing and disruptor Agents, and with feedback credibility check.

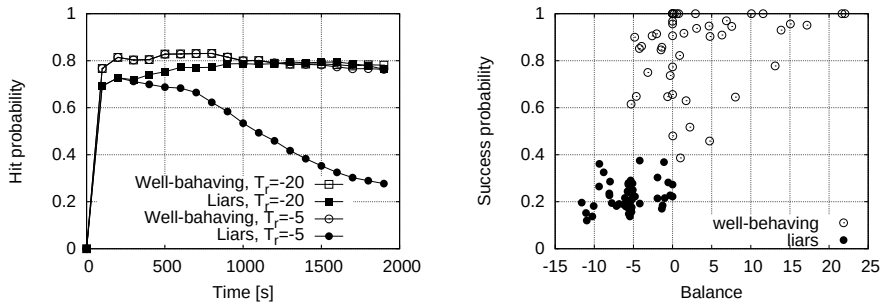
Fig. 5.2(b) shows that, when the banning is enabled, disruptors have lower query success probability *and* lower balance (most of the times below T_r) than well-behaving Agents. Indeed, requests that come from banned Agents are discarded by the Broker but do trigger a balance decrease.

Next, we consider an even more challenging scenario, where 20% Agents are disruptors and other 20% take part in a bad-mouthing attack. Fig. 5.3 shows the time evolution of the query success probability for well-behaving Agents in presence of bad-mouthing attackers, in both the cases where the feedback credibility check (described in Sec. 5.4) is enabled and disabled. Results are compared also with the case where no Agent takes part in the bad-mouthing attack. We first observe that when the credibility check on negative feedbacks is disabled, bad-mouthing Agents slowly but steadily erode into the query success probability of well-behaving users, having them repeatedly banned. Conversely, enabling the credibility check allows well-behaving Agents to achieve the same performance as in the case where no bad-mouthing attack is launched. This behavior highlights two important facts: not only does the credibility check neutralize bad-mouthing (i.e., it has very few false negatives), but it also has very few false positives, i.e., it does not erroneously discard truly negative feedbacks. Indeed, if the feedbacks against real disruptors were discarded, a decrease in the well-behaving Agents' performance would occur, similarly to what is shown in Fig. 5.2(a).

These observations are confirmed by the results in Fig. 5.3(b), detailing the success probability of well-behaving, bad-mouthing and disruptor Agents, versus their balance values. Disruptors still have a lower balance and query success probability than the

Agents (either bad-mouthing or well-behaving) that do provide the content they advertise. Recall that, as explained in Sec. 5.4.1, no action is taken against bad-mouthing Agents, once they are discovered and made harmless.

In addition to disruption and bad-mouthing, there is a further subtle, unfair behavior that Agents may follow: they omit, at registration time, to declare to the Broker which content they wish to share within the Colony, i.e., they pretend they have none. The Broker has no way to find out which content Agents have in their memory, thus these lying Agents will never be selected to provide a content. Also, since there is no evidence of unfair behavior, they will not be banned. However, Figaro effectively tackles this issue: lying Agents do not provide any content and, thus, their balance will soon reach T_r. From then on, they cannot obtain any service, which is the same effect banning would obtain. A similar effect occurs if Agents declare fewer content items than they have.



(a) (b)

Figure 5.4. Resilience to lying Agents: (a) success probability for well-behaving and liars, for $T_r = -20,\ -5$; (b) success probability vs. balance for well-behaving and liars, for $T_r = -5$.

The benefit of setting T_r to a slightly negative value (namely, -5), as opposed to using a larger negative threshold (namely, -20) is evident from Fig. 5.4. When $T_r = -20$, lying Agents achieve almost the same success probability as well-behaving Agents. Conversely, when $T_r = -5$, the balance and success probability of liars severely degrades, i.e., lying about the stored content is not a good choice for rational users.

As a conclusion, a small negative value for T_r makes Figaro highly resilient to both disruption and lying about one's ability to contribute to the Colony.

5.7.3 Cooperation and fairness

We assess the performance of Figaro in terms of fairness, by focusing on the query success probability obtained by the Agents providing the different types of content items,

as listed in Sec. 5.7.1. In particular, we aim at investigating the relationship between balance and query success probability, and how the threshold T_r affects both.

Table 5.1. Query success probability for different content classes and values of T_r

Class	Requester success probability			Provider success probability		
	$T_r = -20$	$T_r = -5$	$T_r = -1$	$T_r = -20$	$T_r = -5$	$T_r = -1$
1	0.910	0.801	0.431	0.937	0.935	0.882
2	0.912	0.820	0.522	0.935	0.922	0.661
3	0.959	0.826	0.532	0.919	0.899	0.441
4	0.966	0.925	0.655	0.904	0.695	0.260

Table 5.1 presents, for each of the four classes of content, the success probability of a query issued by an Agent requesting that content as well as the success probability experienced by an Agent that advertises that content and provides it upon Broker's request. The results refer to three different settings of the request threshold, namely $T_r = -1, -5, -20$. Recall that the higher the content value (i.e., its popularity level and/or size), the higher the number of credits needed to request the content. It follows that the lower the T_r, the more likely it is that an Agent has enough credits to request a content, even if highly valuable, hence the higher its success probability. Consistently, providing a content that is either popular or large-sized, results in a higher gain, hence in better performance for the Agent storing that content.

Next, Fig. 5.5 shows how changing T_r impacts on the relationship between the amount of service (expressed in credits) that Agents provide and obtain from the system. The different markers denote Agents that provide content belonging to different classes. From a fairness viewpoint, we make the following observations. First, in each plot, points lying on the bisectrix $y = x$ correspond to Agents enjoying as much service as the amount they give to the Colony, while points above and below the bisectrix represent Agents that, respectively, obtain and provide more than what they should. Secondly, we would like all Agents to experience the same quality of service, i.e., they can access the

Figure 5.5. Amount of service (in credits) given and obtained by Agents providing different classes of content, when (a) $T_r = -1$, (b) $T_r = -5$ and (c) $T_r = -20$.

same amount of content, independently of what they store.

Looking at the figure, we note that the closer T_r to 0, the more points lie on the bisectrix. However, considering the results in Table 5.1, it is clear that query success probability and fairness are diverging objectives and that properly selecting T_r helps in establishing a tradeoff between the two trends. Specifically, $T_r = -5$ appears to be a good choice, as it both provides high success probability and ensures that each Agent receives about the same amount of service it obtains from the system. However, the content sharing system by itself cannot solve the second issue related to fairness: as is evident from Fig. 5.5, Agents storing low-value items (e.g., class 4) both provide and enjoy little amount of service, with respect to Agents offering more valuable items. We address this problem as described next.

5.8 Exploiting Caching Capabilities

To lessen the effect noted in Fig. 5.5 and increase the content availability in the system (i.e., the query success probability), we enhance our match-making paradigm by letting Agents have caching capabilities, i.e., the possibility to store content items they are not directly interested in, with the sole purpose of helping the Colony (and get a reward for that). They may use such capabilities following the Broker's directions. In other words, some of the Agents that obtain a content can be asked by the Broker to retain it in their cache. Those Agents will then be able to provide the cached content to others.

Below, we formulate our problem and devise a solution that promotes caching Colony-wise. For clarity, we presently leave disruptors and other attackers out of the picture.

5.8.1 Problem formulation

Without loss of generality, we assume that if an Agent has caching capabilities, its cache size is equal to σ. We focus on Colony k and denote by $R_k(j,t)$ the number of Agents in the Colony that advertised content j when they registered, and by $L_k(j,t)$ the number of Agents that are caching content j at time t within the Colony, because asked to do so by the Broker. Note that the latter did not advertise content j during registration, but acquired it in response to a query. Since the two behaviors are mutually exclusive, $P_k(j,t) = R_k(j,t) + L_k(j,t)$.

Let us first compute the query success probability conditioned to the fact that an Agent issuing the query has enough budget to request a desired content. By restricting our attention to well-behaving Agents, this is given by the joint probability of the following events: (i) neither SREQ nor SREP are lost; (ii) in the Colony there is at least one Agent under coverage advertising the requested content. Since it can be assumed that these events are independent and the number of Agents advertising a content does not vary during an SREQ/SREP exchange, the success probability of the query generated

by the generic Agent, for content j in Colony k, is given by

$$S_k(j,t) = (1-q)^2 \left[1-(1-\rho)^{R_k(j,t)+L_k(j,t)}\right]. \tag{5.9}$$

In (5.9), we assumed that SREQ and SREP transfers fail with equal probability q, while `ping` packets (which are very short) are always successfully delivered; ρ is the probability that a generic Agent in the network is under the network coverage.

The following Lemma shows that increasing $S_k(j,t)$ corresponds to increasing the number of Agents providing content j in Colony k.

Lemma 1. *The expression in (5.9) increases monotonically as $R_k(j,t) + L_k(j,t)$ increases.*
The proof is omitted.

However, just increasing the number of Agents providing any content leads to a waste of caching resources. Our caching strategy, instead, needs to adapt the number of copies of the content to the query rate associated to it. To this end, we resort to the content burden metric introduced in (5.1) and define $A_k(t)$ as the number of Agents in the Colony that have caching capabilities (i.e., they can cache some content according to the Broker's directions). Assuming that $A_k(t)$ is known by the Broker and that content popularity is negligibly affected by the change in number of providers due to caching, we are in a position to formulate our goal as minimizing the largest content burden, or, equivalently, as

$$\max \min_j \frac{1}{B_k(j,t)} \tag{5.10}$$

$$s.t. \quad L_k(j,t) \leq A_k(t) - R_k(j,t) \quad \forall j \tag{5.10.1}$$

$$\sum_{j=1}^{I} s(j) L_k(j,t) \leq \sigma \cdot A_k(t) \tag{5.10.2}$$

$$L_k(j,t) \in \mathbf{N} \quad \forall j \tag{5.10.3}$$

Note that such a formulation is an ILP (Integer Linear Programming) problem with I decision variables ($L_k(j,t)$, $j = 1,\ldots,I$). Constraint (5.10.1) forces the number of cached copies for content j to be not greater than the number of caching-capable Agents not advertising that content, while Constraint (5.10.2) ensures that the total number of cached items does not exceed the cache capacity available in the system. Constraint (5.10.3) forces the decision variables to take non-negative integer values. Unfortunately, a polynomial or pseudo-polynomial time solution to the above problem does not exist [61]. Additionally, the system dynamics require the problem to be solved every time Agents enter/leave the system, or they are banned/unbanned ($R_k(j,t)$ changes), or if the content of an Agent's cache is modified ($L_k(j,t)$ changes). Thus, we devise a heuristic to handle the problem solution, and evaluate its performance in Sec. 5.9.1.

5.8.2 A heuristic caching strategy

Our heuristic is implemented at the Brokers, which are in a good position to classify content based on its rarity and popularity, since they know the number of providers and have a running estimate of the query rate for any content within their Colony. Also, using SREP messages, they can ask the Agents that are retrieving a content item to add it to their cache. Feedback packets can be used by Agents to inform the Broker whether they followed its directions and which content, if any, they discarded to make room for the new content. In this way Brokers complement their knowledge of the number $R_k(j,t)$ of initial providers advertising the content, with that of the number $L_k(j,t)$ of caching Agents.

The Broker considers that a content j is worth to be cached (hereinafter referred to as *cacheworthy*) if the content burden, $B_k(j,t)$, outweighs the average value (computed over all content items available in the Colony) by a factor $\phi > 1$. As an Agent in the Colony issues a query for a cacheworthy content, the Broker asks the Agent to cache it if its balance is smaller or equal to the average. Also, it returns to the Agent the burden of the content items, in order to provide a discard priority for different items in case of cache overflow (i.e., content associated to lower burden is more likely to be discarded).

From the Agent's viewpoint, we define the benefit/cost ratio of providing a content j as $\frac{G_k(j,t)}{c(j)} = \frac{g+B_k(j,t)}{K}$. This metric represents how much an Agent's balance increases per unit of effort (i.e., for a unitary amount of transferred data). Then, we prove that rational Agents will follow the Broker's suggestion to cache a content, whenever such a content is cacheworthy.

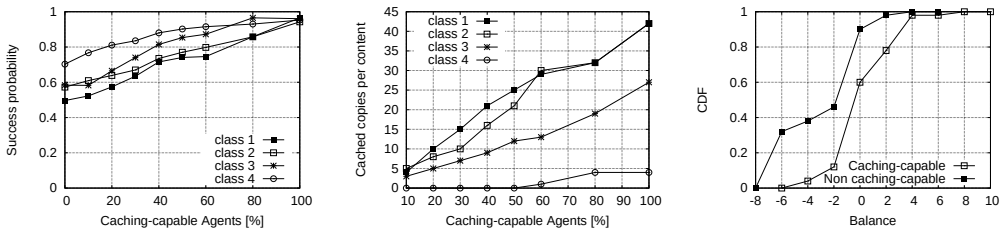
Theorem 5.8.1. *Given a rational Agent currently storing the set of content items $\mathcal{S}$, the Agent always finds it convenient to cache a new content w, upon Broker's suggestion.*

Proof. The proof is omitted. □

Based on the above theorem, we conclude that the proposed heuristic can be successfully implemented: rational Agents will follow the Broker's suggestions, as it allows them to become providers of a content with higher benefit/cost ratio and, thus, increase their expected reward.

5.9 Evaluating Figaro with Caching

We now evaluate the effectiveness of Figaro's caching mechanism in improving the Agents' query success probability and mitigating the unfairness among Agents providing items with different characteristics. We also compare the performance of Figaro against other existing solutions.



(a) (b) (c)

Figure 5.6. (a) Query success probability and (b) number of cached copies for the different content classes and as the percentage of Agents with caching capability varies; (c) CDF of the balance for Agents able/unable to cache.

5.9.1 Simulation Results

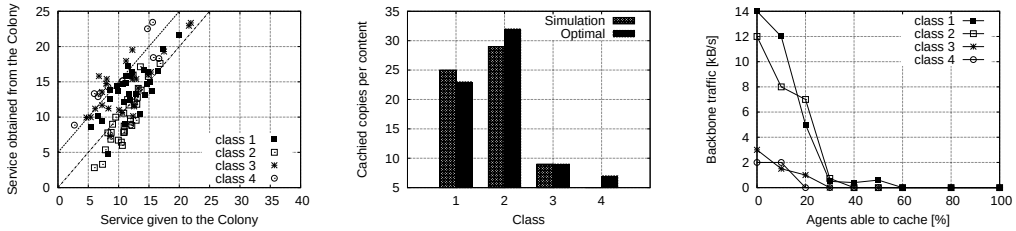
We now consider that a certain percentage of Agents have caching capabilities, and, for clarity of presentation, that there are no disruptors. We want to address the following questions:

- is caching effective in improving Figaro's performance?
- how many caching-capable Agents are needed for caching to work?
- how does caching impact on the balance distribution in the Colony?

To this end, we set $T_r = -5$, $\sigma = 200$ kB, and $\phi = 1.5$, and show the performance of the proposed heuristic caching strategy as the percentage of caching-capable Agents varies.

Figs. 5.6(a) and 5.6(b) show, respectively, the query success probability and the number of copies cached in the system, for the different content classes. As expected, as the number of caching-capable Agents increases, the success probability increases as well. However, a query for highly popular, large-sized content (class 1) has lower chances to succeed than others, unless all Agents can cache additional content items. Indeed, less requested or smaller items are less valuable (in terms of credits), hence they are easier to obtain. Nevertheless, Figaro significantly reduces the query success probability gap between the different content classes, already with 40% caching-capable Agents. Interestingly, Fig. 5.6(b) shows that what matters for a content to be cacheworthy is mostly its popularity level: most of the cached content items are the popular ones (i.e., classes 1 and 2). However, as the popular items become widely available in the system, room can be devoted to less requested items, especially the large-sized ones (i.e., class 3).

Next, we fix the percentage of caching-capable Agents to 50% and present in Fig. 5.6(c) the balance cumulative distribution function (CDF), for caching-capable and not caching-capable Agents. Caching-capable Agents have a significantly higher balance, due to the



(a) (b) (c)

Figure 5.7. (a) Service given and obtained by Agents and (b) heuristic caching strategy vs. optimal solution, for 50% caching-capable Agents; (c) Backbone usage as the percentage of caching-capable Agents varies.

higher burden (hence value) of the content they provide. Also, notice that with extremely high probability caching-capable Agents have a balance greater than T_r, showing that caching has also the positive effect of distributing more evenly the load among Agents.

This is confirmed by Fig. 5.7(a): caching improves not only the query success probability, but also the system fairness. In contrast to Fig. 5.5b, the amount of service provided and obtained by Agents does not depend any longer on the class of the content Agents originally advertised: *all* Agents receive from the Colony a level of service that is close to the one they provide *and*, very likely, they have a balance greater than the threshold T_r. In other words, caching is an effective way to achieve *both* a very high success probability and fairness guarantees among the Agents.

Next, for the different content classes, we compare the number of copies cached in the system as obtained through our heuristic (implemented in simulation), with the solution to the optimization problem in (5.10). The latter is computed by assuming that: (i) conditions are stationary (i.e., Agents are static with probability $\rho = 0.75$ to be under coverage, which is in agreement with the network scenario under study), and (ii) the Broker has knowledge of the number of caching-capable Agents as well as of the status of all Agent caches. The agreement between the results, shown in Fig. 5.7(b), proves the good performance of Figaro, even compared to the case where global knowledge is assumed at the Broker.

Finally, caching also has the positive effect of reducing the usage of the backbone. Indeed, caching increases the content availability inside the Colony and, consequently, reduces the need to search for it outside (i.e., asking the Proxy). Fig. 5.7(c) confirms this statement, and suggests that, when used, the backbone is mainly employed to retrieve popular content. This is of particular importance when the backbone is not wired but implemented with cellular technologies such as 3G. In those cases, reducing its usage results in significant monetary savings, and may represent a strong motivation to deploy a peer-to-peer content discovery system like Figaro.

5.9.2 Benchmarking Figaro

We finally evaluate Figaro by comparing it with other schemes in terms of query success probability and overhead. In Figaro, we assume that 50% of the Agents have caching capabilities. Since the results above showed that Figaro with caching greatly mitigates the differences in performance among Agents advertising content items with different characteristics, we now show results averaged over the different content classes.

Figaro is compared with a simple content-retrieval mechanism, referred to as Flat Flooding, and the BubbleStorm scheme, adapted to our wireless scenario from its wireline version [57].

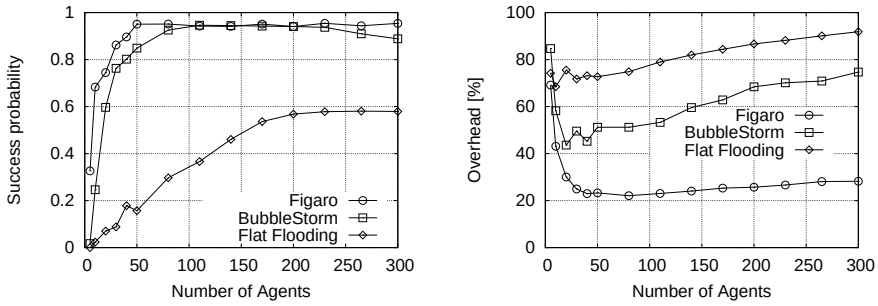
Flat Flooding hinges on a flat peer-to-peer exchange in ad hoc mode connectivity (i.e., without infrastructure). The query propagation range is spatially limited by a Time To Live value (set to 10 hops), and the rebroadcasting of already solved queries is avoided by means of a query lag time (set to 1 s): if an Agent detects responses to a query it has just received, it refrains from forwarding it any further. In this infrastructure-less scenario, Agents implement a routing protocol for ad hoc networks (we chose OLSR) and act as relay nodes when needed.

As mentioned in Sec. 5.2, BubbleStorm [57] is based on a probabilistic exhaustive search paradigm in a overlay network. In our scenario, given the topological constraints, the random graph structure used by BubbleStorm to propagate a content is built as a subset of the tree already provided by the infrastructure. Also, our implementation of BubbleStorm provides for nodes to replicate, broadcast and cache content in such a way that the average fraction of Agents storing a content in BubbleStorm is equal to the one set up for Figaro.

Notice that, while deriving the results, in Figaro Agents are assumed to be rational, i.e., to cooperate only if it is beneficial to them; in the case of the other schemes, instead, an ideal behavior is assumed, i.e., Agents are always willing to cooperate.

The query success probability is reported in Fig. 5.8(a), as the number of Agents varies. We note that Figaro outperforms the other solutions, especially when the number of Agents is low. While the comparison against Flat Flooding (which does not exploit any infrastructure) is not surprising, the improvement with respect to BubbleStorm is less obvious. Indeed, in the wireless scenario under study, BubbleStorm performs slightly worse than its wireline version, whose success probability exceeds 0.99. Figaro, instead, is more suitable for wireless, dynamic scenarios due to the match-making capabilities of the Brokers. The role of the Broker in arbitrating content sharing within a Colony also explains why the performance of Figaro does not deteriorate as the number of Agents grows, as instead happens when BubbleStorm's statistical matching is used.

An area where Figaro provides performance unmatched by BubbleStorm is message overhead. Fig. 5.8(b) shows that Figaro exhibits an overhead that is nearly inversely proportional to the query success probability, as a higher success probability implies that very likely (i) a query is satisfied within the Colony where it has been generated



(a) (b)

Figure 5.8. (a) Success probability averaged over the different items and (b) message overhead, as functions of the number of Agents.

and (ii) fewer SREQs are issued. In particular, it is shown that the Figaro overhead stabilizes below 30% of the total traffic. BubbleStorm instead exhibits a significantly higher overhead, due to its proactive content (and query) propagation over the network. Finally, with Flat Flooding the overhead increases as the number of Agents grows, due to the increased network congestion: large numbers of Agents trigger an overwhelming number of replies to a single query.

5.10 Conclusions

We presented Figaro, a match-making content discovery solution for wireless networks with infrastructure. In Figaro, mobile users (a.k.a. Agents) provide and request content items, while fixed Brokers help Agents in identifying who owns a desired content. A balance system ensures that Agents are treated in a fair way, i.e., the amount of service they receive is comparable to the amount of service they provide. Contributing to the effectiveness of Figaro is its feedback mechanism, that allows Brokers to zero in on free riders as well as attackers, and ban them to limit their negative impact. Also, Figaro complements its design with a caching scheme in which the Brokers suggest to the Agents what content to cache, in order to increase the query success probability (global and Agent's) and ensure fairness among Agents. Finally, we proved that it is rationally convenient for Agents to cooperate and to follow the Broker's caching advice. Simulation results showed the resilience of Figaro against different types of attacks, as well as its effectiveness, also with respect to other existing solutions.

Chapter 6

Conclusions and future work

In this thesis, we discussed the planning and management of mobile networks with infrastructure.

We started by addressing the problem of infrastructure planning. Assuming a perfect knowledge of vehicular mobility, we developed a graph model able to capture the evolution of the network topology, as well as to account for channel access issues. By solving a set of max-flow problems over such a graph, we were able to conclude that popular placement heuristics such as placing the APs in the most crowded locations are significantly suboptimal. Furthermore, we could observe that most data flow through vehicle-to-vehicle links, but only a negligible fraction travel more than two hops. This in turn suggests that the effort of building a complex multi-hop protocol can be avoided in certain scenarios. Finally, we found that data can travel over significant distances through carry-and-forward.

We extended the scope of our work in order to account for the fact that out knowledge of vehicular mobility is affected by several kinds of error. We enhanced our graph model in order to include probabilistic edge weights, and studied the effect of prediction inaccuracy over the network performance. We found that the most significant effect of prediction inaccuracy is not a lower throughput, but rather a higher amount of data unnecessarily sent from APs to vehicles they uncorrectly believe to be potential relays. We also addressed the case in which contents are location-specific, finding that such a scenario is more sensible to the effects of prediction inaccuracy. As a complement to our modeling effort, we verified that a simple, second-order Markovian prediction technique can be accurately described by our model, which can effectively be used to study its performance.

Then, we switched to a non-cooperative scenario, in which the AP deployment is not decided in a centralized way, but is the result of the action of several competing operators. For simplicity, we restricted our attention to a scenario with two possible AP locations, and two operators playing a leader-follower game. We found that whether the follower choses to place its AP in the same location of the leader depends in a non-trivial

way upon the vehicular flows being dealt with, as well as the content size. Finally, if the follower can freely decide the distance between its AP and the leader's one, it will choose a distance that makes the coverage areas partially overlap.

As far as network management is cncerned, we presented a content-discoovery solution based on a variant of the publish-and-subscribe paradigm. By combining a balance system and a feedback and banishment mechanism, we were able to ensure that rational (i.e., self-interested) users always provide the content when requested by the Broker, while malicious users are effectively detected and isolated. By allowing those Agents with caching capabilities to use them following the Broker's suggestions, we were also able to increase the availability of popular and/or rare contents.

Future work will focus on two trends. On one hand, we will investigate the potential benefit of including parked vehicles in the content delivery process. On the other, we will look at how our graph-based model can be integrated in joint spectrum and AP location auctions.

Appendix A

Proactive Seeding for Information Cascades in Cellular Networks

A.1 Introduction

Cellular traffic is growing exponentially, tripling every year, with a share of video traffic increasing from 50% now to an expected 66% by 2015 [67]. Credit Suisse reported in [66] that 23% of base stations globally have utilization rates of more than 80 to 85% in busy hours, up from 20% last year.This dramatic increase in demand is generating serious problems for 3G networks and these problems are likely to remain in 4G networks as well. Another aggravating fact for the operators is that the cellular network traffic greatly fluctuates throughout the day, following strong daily and weekly patterns, as we show in Fig. A.4(c). Since the cellular network is provisioned for *peak traffic*, any capability that can distribute the network load more evenly over time would significantly address the current as well as future capacity shortcomings for the operator.

At the same time, in today's Internet, online social networks (OSNs)[1] are becoming an increasingly important way in which users are informed about content. This is not surprising: people tend to value highly the content recommended by friends or people with similar interests (e.g., members of the same groups), and are also likely to recommend it further to others.

The growth of cellular traffic and of OSN's importance are inherently related. Indeed, mobile devices are quickly becoming the primary mean to access OSNs. For example, one third of all Facebook users regularly access the service from their mobile devices and they generate twice as much activity than non-mobile users [62]. Consequently, the interest diffusion over OSNs translates directly into increased cellular

[1]By OSNs here we refer to online social networks such as Facebook and Twitter, websites with social networking features such as Digg.com, blogs, email communication, and other online networks that exploit social ties for interest diffusion.

traffic.

Cellular operators may try to exploit the knowledge of such interest diffusion to alleviate the peak demand in cellular traffic. One approach is to *delay* some of the traffic, e.g., by limiting the diffusion of interest [63] or by using techniques that trade-off user delay for traffic load [64, 65].

We take a different approach and aim at serving *impatient* users, i.e., users that expect the content right after they demand it and do not tolerate large delays and jitters. Our key observation is that given the vast information often available to the cellular operator (e.g., address books, session logs, location history, partnerships with OSNs, etc.), the network can detect information cascades and predict the future demand. Consider, for example, the case of Youtube videos: Google reported that up to 200 million Youtube videos per day were delivered to mobile devices in 2010 [67]. Many views of these videos are due to the spread of their URLs over various OSNs. The evolution of such cascades of forwarded URLs depends on the structure of the OSN, similarity of users and other relevant features. With this information, it is possible to model and predict the diffusion of interest [68, 69]. For example, in [70], the authors apply machine learning techniques to Twitter traces, and predict more than half of URL-based cascades of tweets with only a 15% false positive rate.

In this chapter, we propose Proactive Seeding, a technique for reducing the peak load in cellular networks, while providing users with low (or zero) access latency. Proactive Seeding exploits the predictability of future demand by proactively pushing ("seeding") the content to users before, and no later than, they request it. This allows us to move some cellular traffic from the busiest hours to times with lower load and thus to reduce its peaks, as illustrated in Fig. A.1. Proactive Seeding is optimal in the offline setting (i.e., assuming perfect knowledge of all information cascades), in the sense that it minimizes the peak load while delivering the content to a user no later than she requests it. In our simulation driven by traces from Twitter and cellular networks, Proactive Seeding leads to 20%-50% reduction in the cellular peak load. In the case of imperfect prediction, where the gains are naturally reduced, we show that the conservative approach of underestimating the future demand still guarantees positive gains. Finally, we show how Proactive Seeding can be combined with techniques [71, 72] that exploit the local device-to-device (D2D) connectivity (over WiFi or Bluetooth), and that such a combination performs better than each technique separately.

The structure of the rest of the chapter is as follows. In Section A.2, we provide the formal problem statement. In Section A.3, we present the Proactive Seeding solutions under the assumption that demand can be perfectly predicted. In Section A.4, we modify our framework to allow for imperfect, probabilistic estimation of the prediction. In Section A.5, we present our evaluations results. We overview the related literature in Section A.6, then conclude the chapter with Section A.7.

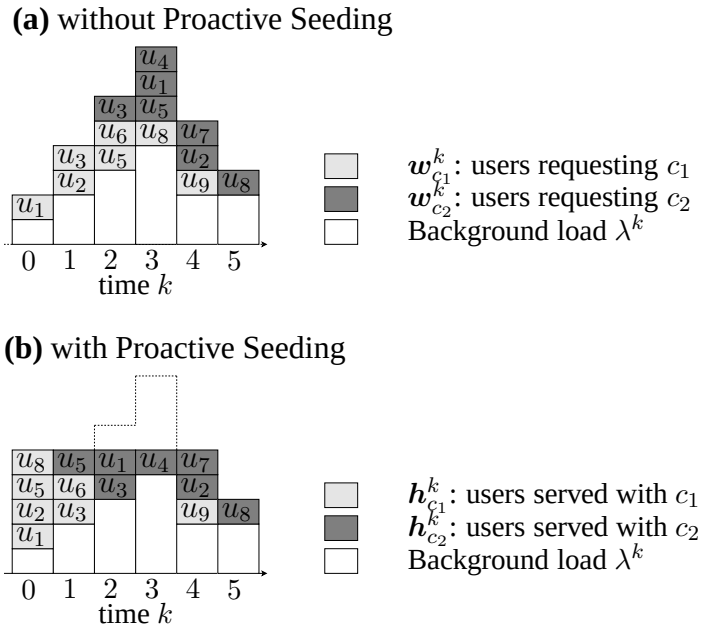


Figure A.1. Illustration of Proactive Seeding in a system with two types of contents $\boldsymbol{C} = \{c_1,c_2\}$ disseminated among 9 users $\boldsymbol{U} = \{u_1 \dots u_9\}$, in presence of the background load λ^k. **(a)** The diffusion of interest between the users in content c_1 (bright gray) and c_2 (dark gray). For example, $u_3 \in \boldsymbol{w}^2_{c_2}$ means that user u_3 becomes interested in content c_2 at time $k=2$. Without Proactive Seeding, users request and pull the content through cellular right when they get interested in it ($\boldsymbol{h}^k_c \equiv \boldsymbol{w}^k_c$), which results in an uneven total cellular load (the total height of bars). **(b)** Proactive Seeding serves some users before they actually become interested in the content ($\boldsymbol{W}^k_c \subseteq \boldsymbol{H}^k_c$). The total load becomes more even in time and its peaks decrease (here by 3 units).

A.2 Problem statement

We distinguish between two components of cellular traffic: (i) background load and (ii) predictable traffic.

A.2.1 Background cellular load

We refer as background (cellular) load to all traffic which is out of our control: its content cannot be predicted (at least not with a reasonable accuracy) and/or served before the actual request occurs. For example, phone conversations and other types of real-time traffic contribute to background load. We denote by λ^k the total amount of background load at time frame k, $0 \leq k \leq K$.

We illustrate λ^k by white bars in Fig. A.1; note that because the content composing it cannot be predicted or served earlier, λ^k remains unchanged in Fig. A.1(b).

A.2.2 Predictable cellular traffic

In contrast, the predictable cellular traffic is all the traffic that can somehow be predicted and thus proactively served. Denote by $\boldsymbol{U}$ the set of all users, and by $\boldsymbol{C}$ the set of all existing pieces of predictable content. We assume that transmitting a single piece $c \in \boldsymbol{C}$ of content to a single user $u \in \boldsymbol{U}$ takes exactly a single unit of cellular traffic.[2] Now, denote by $\boldsymbol{w}_c^k \subset \boldsymbol{U}$ the set of users that demand ("want") the content $c \in \boldsymbol{C}$ exactly at time frame k. In other words, $\boldsymbol{w}_c^k$ describes the diffusion of interest in content c (typically over OSNs). Let

$$\boldsymbol{W}_c^k = \bigcup_{m=0}^{k} \boldsymbol{w}_c^m \quad (\boldsymbol{W}_c^k \subseteq \boldsymbol{U}) \tag{A.1}$$

be the cumulative version of $\boldsymbol{w}_c^k$, i.e., the set of all users that have requested c until frame k. Finally, we denote by $k(u,c)$ the time when user u demands content c, i.e., such that $u \in \boldsymbol{w}_c^{k(u,c)}$.

In the example in Fig. A.1(a), $\boldsymbol{w}_{c_1}^2 = \{u_5,u_6\}$ and, consequently, $k(u_5,c_1) = k(u_6,c_1) = 2$.

A.2.3 Transmission schedule

In this chapter, we decouple the diffusion of interest in the content (i.e., demand) from the actual delivery process. To this end, we denote by $\boldsymbol{h}_c^k \subset \boldsymbol{U}$ the set of users that get ("have") content c over cellular network exactly at frame k. Its cumulative version

$$\boldsymbol{H}_c^k = \bigcup_{m=0}^{k} \boldsymbol{h}_c^m \qquad (\boldsymbol{H}_c^k \subseteq \boldsymbol{U})$$

is the set of all users that have c at frame k. In the other words, $\boldsymbol{h}_c^k$ is a *schedule* that determines when the cellular operator sends content c to which users.

For example, in Fig. A.1(b), $\boldsymbol{h}_{c_1}^1 = \{u_3,u_6\}$ and $\boldsymbol{h}_{c_2}^1 = \{u_5\}$.

A.2.4 User Impatience

In this work, we consider the case where all users are *impatient*: a user $u \in \boldsymbol{U}$ wants to enjoy content $c \in \boldsymbol{C}$ right after she becomes interested in it. This means that u should

[2] In practice, the content spread over OSNs may greatly vary in size: a ten-minutes-long Youtube movie is orders of magnitude bigger than a photograph. All the equations can be easily modified to reflect heterogeneous content size, at the cost of notation clarity.

receive c at time l not larger than $k(u,c)$, i.e., $u \in \boldsymbol{h}_c^l$ such that $l \leq k(u,c)$. This is achieved by guaranteeing that

$$\boldsymbol{W}_c^k \subseteq \boldsymbol{H}_c^k \quad \text{for every } k \text{ and } c. \tag{A.2}$$

For example, in Fig. A.1(b), we push content c_1 to user u_5 at time $k=0 < k(u_5,c_1) = 2$, which is allowed by (A.2). In contrast, sending it at time $k > 2 = k(u_5,c_1)$ would violate the constraint in (A.2).

A.2.5 Objective

Using the notation above, the *total cellular traffic/load* at time k can be decomposed as the sum of background cellular load and total predictable traffic, i.e.,

$$\text{total cellular load} \;=\; \lambda^k + \sum_{c \in \boldsymbol{C}} |\boldsymbol{h}_c^k|. \tag{A.3}$$

Our objective is to minimize the peak of total cellular load, i.e.,

$$\text{minimize} \quad \max_{0 \leq k \leq K} \left(\lambda^k + \sum_{c \in \boldsymbol{C}} |\boldsymbol{h}_c^k| \right) \tag{A.4}$$

subject to the user impatience constraint in (A.2).

Note that because we have no control over the diffusion of interest $\boldsymbol{w}_c^k$, we can affect (A.4) only by choosing the schedule $\boldsymbol{h}_c^k$. We give an example of such an optimized schedule in Fig. A.1(b). In particular, we (i.e., the cellular operator) predict which users will be interested in content c, and proactively *seed* some of them with c when the cellular load is relatively small, e.g., during the previous night. This allows us to reshape the cellular traffic and reduce its peaks, but not the total traffic.

A.3 Proactive Seeding Algorithms

In this section, we focus on the *offline* case, where we have perfect knowledge of the future diffusion of interest, i.e., we know $\boldsymbol{w}_c^k$ for all time frames k and pieces of content c. The offline case serves as a baseline for understanding the maximum achievable gains. It also serves as a building block for the more realistic, *online* scenario, where prediction of the future is imperfect, described in Sec. A.4.

A.3.1 Special Case: single content, no background load

Let us first consider the simplest, yet intuitive case: there is only a single content ($\boldsymbol{C} = \{c\}$) and no background load ($\lambda^k = 0$). An example of the demand curve corresponding to such a cascade (e.g., a single content flash-crowd) is shown in Fig. A.2: the

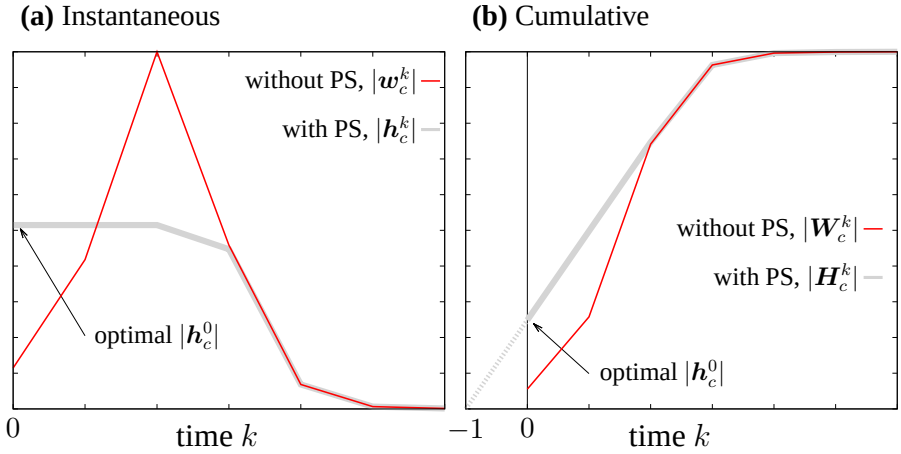


Figure A.2. Geometric interpretation of optimal Proactive Seeding (PS) under a single content cascade ($\boldsymbol{C} = \{c\}$), with no background cellular load ($\lambda^k = 0$), as described in Sec. A.3.1. The curve represents a typical cascade on the Facebook social graph (see Sec. A.5.3). We minimize the peak instantaneous cellular load in **(a)** while satisfying the impatience constraint (A.2), by proactively seeding the users at a constant rate, until the cascade passes. The optimal seeding rate $|\boldsymbol{h}_c^0|$ can be found by studying the cumulative version **(b)** of the time evolution, where a line anchored at point (-1,0) and tangential to $|\boldsymbol{W}_c^k|$, crosses the y-axis at point (0,$|\boldsymbol{h}_c^0|$).

total number of users interested in the content increases until reaches a peak and then decreases.

In this special case, objective (A.4) is equivalent to minimizing $\max_k(|\boldsymbol{h}_c^k|)$ subject to the user impatience constraint (A.2). Intuitively, this entails delivering the content more evenly over time. Ideally, we would like to send the content with a constant *seeding rate* $|\boldsymbol{h}_c^k|$ and thus at linear $|\boldsymbol{H}_c^k|$.This rate should be the lowest possible, while still satisfying (A.2). Because $\boldsymbol{C} = \{c\}$, (A.2) is satisfied if $|\boldsymbol{W}_c^k| \leq |\boldsymbol{H}_c^k|$ for every k. Consequently, $|\boldsymbol{H}_c^k|$ should be linear and never smaller than $|\boldsymbol{W}_c^k|$. This leads to an intuitive geometric solution: Draw a straight line that crosses point (-1,0) and is tangential to $|\boldsymbol{W}_c^k|$. The optimal service rate $|\boldsymbol{h}_c^k|$ is determined by the point where the line crosses the y-axis. We show an example in Fig. A.2.

It is also easy to see that this optimal rate $|\boldsymbol{h}_c^k|$ is also provided by the following formula

$$|\boldsymbol{h}_c^k| \;=\; \left\lceil \max_{l=k}^{K} \frac{|\boldsymbol{W}_c^0| - |\boldsymbol{H}_c^l|}{l+1} \right\rceil . \tag{A.5}$$

A.3.2 General Case: multiple contents, background traffic

The simple geometric solution from Sec. A.3.1 does not directly extend to the general case, i.e., in presence of arbitrary background cellular load $\lambda^k > 0$ and multiple contents $|\boldsymbol{C}| > 1$. For example, (A.5) would not necessarily satisfy the user impatience

Algorithm 1 Proactive Seeding

Require: $\boldsymbol{w}_c^k \; \forall c,k, \quad \lambda^k \; \forall k$ *future demand and load*

1: $\boldsymbol{h}_c^k \leftarrow \emptyset \quad \forall c,k$
2: $L \leftarrow \emptyset$
3: **for all** (u,c) such that $u \in \boldsymbol{W}_c^K$ **do**
4: $\quad L \leftarrow L \cup \{(u,c)\}$
5: **end for**
6: **sort** L by increasing $k(u,c)$
7: **for all** (u,c) in L **do** *water-filling*
8: $\quad k^* \leftarrow \arg\min_{0 \le l \le k(u,c)} (\lambda^l + \sum_c |\boldsymbol{h}_c^l|)$
9: $\quad \boldsymbol{h}_c^{k^*} \leftarrow \boldsymbol{h}_c^{k^*} \cup \{u\}$
10: **end for**
11: **return** $\boldsymbol{h}_c^k \quad \forall c,k$ *optimal*

constraint (A.2) for each of the $|\boldsymbol{C}| > 1$ contents separately.

To address these problems, we propose the Proactive Seeding algorithm, shown in Alg. 1. We construct the seeding schedule $\boldsymbol{h}_c^k$ iteratively, starting from an empty set (line 1). In lines 2-6, we create a list L of existing user-content pairs (u,c), sorted according to the growing want times $k(u,c)$. Note that user u may appear in L multiple times, i.e., exactly once for each content c she is interested in. Lines 7-9 implement a water-filling type of algorithm, where for each pair (u,c) we find the time frame $k^* \le k(u,c)$ with the smallest total cellular load $\lambda^{k^*} + \sum_c |\boldsymbol{h}_c^{k^*}|$. We then schedule this pair (u,c) at time k^* by adding u to $\boldsymbol{h}_c^{k^*}$ (line 9). Finally, once all existing pairs (u,c) are scheduled, Proactive Seeding returns the seeding schedule $\boldsymbol{h}_c^k$ for all contents c and time frames k.

We illustrate the output of Proactive Seeding in the example of Fig. A.1(b). The sorted list L resulting after line 6 is $L = [(u_1,c_1),(u_2,c_1),(u_3,c_1),(u_5,c_1),(u_6,c_1),(u_3,c_2),(u_8,c_1),$ $(u_5,c_2),(u_1,c_2),(u_4,c_2),(u_9,c_1),(u_2,c_2),(u_7,c_2),(u_8,c_2)]$. For pair (u_1,c_1), we have $k(u_1,c_1) = 0$, and therefore lines 8-9 result in $k^* = 0$ and $\boldsymbol{h}_{c_1}^0 = \{u_1\}$, respectively. When processing the second element in L, (u_2,c_1), we have $\lambda^l + \sum_c |\boldsymbol{h}_c^l| = 2$ for both $l = 0$ and $l = 1$. We arbitrarily break this tie by setting $k^* = 0$, which results in $\boldsymbol{h}_{c_1}^0 = \{u_1,u_2\}$. The third pair (u_3,c_1) has now a unique $k^* = 1$, and is scheduled therein. The process continues until L is exhausted.

This schedule $\boldsymbol{h}_c^k$ returned by Proactive Seeding is optimal:

Theorem A.3.1 (Optimality of Proactive Seeding)**.** *The seeding schedule $\boldsymbol{h}_c^k, \forall c,k$, created by Proactive Seeding minimizes the peak load (objective in (A.4)), while satisfying the user impatience constraint (A.2) for each content c separately.*

Proof. First, note that the frame k^* chosen for user u in line 8 is not greater than the time $k(u,c)$ when u actually wants the content. Therefore, by construction, the schedule

created by Proactive Seeding always satisfies the user impatience constraint (A.2) for every content c separately.

We now have to prove that the objective (A.4) is met by Proactive Seeding. Denote by $L(j)$ the set of all pairs (u,c) such that $k(u,c) = j$ and by $L(i,j) = \bigcup_{m=i}^{j} L(m)$. Denote by $\boldsymbol{h}(j)$ the transmission schedule constructed by Proactive Seeding just after processing the pairs $L(j)$ in lines 7-9. In other words, $\boldsymbol{h}(j)$ schedules all contents for all users that want it not later than at time j. Consequently, $\boldsymbol{h}(K)$ denotes the entire schedule, $\boldsymbol{h}(K) \equiv \bigcup_{c,k} \boldsymbol{h}_c^k$. We prove the optimality of Proactive Seeding by induction on j, as follows.

Initialization ($j = 0$): For every pair $(u,c) \in L(0)$, line 8 automatically sets $k^* = 0$. Consequently, $\boldsymbol{h}(0)$ schedules all pairs $L(0)$ at time slot 0. This is the only feasible solution, thus the optimal one.

Induction step: Assume that $\boldsymbol{h}(j)$ is optimal for all pairs $L(0,j)$. We now must prove that $\boldsymbol{h}(j+1)$ is optimal for all pairs $L(0,j+1)$.

Denote by $\max(\boldsymbol{h}(j))$ the peak total cellular load resulting from $\boldsymbol{h}(j)$. Either an optimal allocation will increase the peak rate at $j+1$, or keep it constant. Thus we can distinguish two cases, as follows:

Case 1: It is possible to schedule the pairs $L(j+1)$ such that $\max(\boldsymbol{h}(j+1)) = \max(\boldsymbol{h}(j))$. In this case, lines 7-9 guarantee that this equality holds under Proactive Seeding, by iteratively choosing the least loaded time slots. Now, because $\max(\boldsymbol{h}(j))$ is optimal, it is the smallest value that does not violate the impatience constraint (A.2). So $\boldsymbol{h}(j+1)$ cannot be lower than $\max(\boldsymbol{h}(j))$ without violating (A.2). Consequently, $\max(\boldsymbol{h}(j+1)) = \max(\boldsymbol{h}(j))$ implies the optimality of $\boldsymbol{h}(j+1)$.

Case 2: It is *not* possible to schedule the pairs $L(j+1)$ such that $\max(\boldsymbol{h}(j+1)) = \max(\boldsymbol{h}(j))$. We can now distinguish two sub-cases, depending of the background load at time $j+1$:

Case 2.1: If $\max(\boldsymbol{h}(j+1)) = \lambda^{j+1}$ is achievable, then lines 7-9 of Proactive Seeding will achieve that by iteratively choosing the least loaded time slots. In this case, the peak load is equal to the background load λ^{j+1}. Such a peak load is optimal, because, by definition, background load cannot be changed.

Case 2.2: If $\max(\boldsymbol{h}(j+1)) = \lambda^{j+1}$ is *not* achievable, then lines 7-9 guarantee that $\max(\boldsymbol{h}(j+1)) - \min(\boldsymbol{h}(j+1)) \leq 1$, where $\min(\boldsymbol{h}())$ denotes the minimal total cellular load resulting from $\boldsymbol{h}()$. Consequently, $\max(\boldsymbol{h}(j+1))$ cannot be decreased and $\boldsymbol{h}(j+1)$ is thus optimal. □

Note: Although optimal in the sense of objective (A.4), Proactive Seeding does not guarantee that the users will be served in the order they request the content; it may schedule user u before user w, even if $k(u,c) > k(w,c)$. For example, in Fig. A.1 user u_3 wants content c_1 before user u_5, but is scheduled to receive it after u_5, as we show in Fig. A.1(b). However, it is easy to see that an additional step that reshuffles

the users to enforce the "first-want-first-serve" (i.e., chronological) order, preserves the optimality and feasibility of the resulting schedule $\boldsymbol{h}_c^k$.

A.3.3 Extension: D2D-aware Proactive Seeding

In addition to their cellular connections, it is often the case that some users are within physical proximity of each other and can establish direct device-to-device (or D2D [73]) connections between them, e.g., via ad-hoc 802.11 or Bluetooth. If these users are interested in the same content, they can exploit their D2D connectivity, and thus offload the cellular network. Several variants of this idea have been studied in the past, e.g., in [71, 72, 74, 75]. What makes this particularly promising, in our context, is the fact that there is a correlation between geographical proximity and proximity on the social graph [76]. We show below (and later, in simulations) that these techniques can be combined with Proactive Seeding, and address two complementary aspects: using the D2D connections helps to offload the total aggregated cellular load, while Proactive Seeding helps to smooth the load over time.

The D2D connectivity graph changes over time. We denote by $\boldsymbol{N}^k(u)$ all D2D neighbors of user u at time k. Consider time $k(u,c)$ when user u becomes interested in content c. We will assume that each mobile user behaves as follows:

1. If u has been seeded with c before, no action is needed.
2. Otherwise, u attempts to pull c from its current local neighbors $\boldsymbol{N}^{k(u,c)}(u)$. This is possible only if at least one of these neighbors has c, i.e., if $\boldsymbol{N}^{k(u,c)}(u) \cap \boldsymbol{H}_c^{k(u,c)} \neq \emptyset$.
3. Otherwise, u fetches c through the cellular network.

Depending on the extent to which the operator is aware of D2D connectivity, different optimizations are possible:

D2D-unaware Proactive Seeding

In this simplest scenario, the operator does not have information about the location of users and thus performs Proactive Seeding without taking proximity into account. Consequently, user u can benefit from D2D, in an opportunistic way, i.e., only if u has not been seeded earlier (i.e., if $u \in \boldsymbol{h}_c^{k(u,c)} \cap \boldsymbol{w}_c^{k(u,c)}$), which results in

$$\boldsymbol{h}_c^k \leftarrow \boldsymbol{h}_c^k \setminus \left\{ u \in \boldsymbol{h}_c^k \cap \boldsymbol{w}_c^k : \ \boldsymbol{N}^{k(u,c)}(u) \cap \boldsymbol{H}_c^{k(u,c)} \neq \emptyset \right\}.$$

In the example of Fig. A.1, user u_4 will pull content c_2 from its D2D neighbors $\boldsymbol{N}^3(u_4)$ at time $k = 3$ if at least one of them is in $\{u_1,u_3,u_5\} = \boldsymbol{H}_{c_2}^2$ (i.e., already has c_2).

D2D-aware Proactive Seeding

In this scenario, the operator has information about location and thus proximity of users[3] and takes it into account while seeding. In particular, it applies Proactive Seeding but avoids seeding user u if u will be able to get the content from its neighbors. This can be achieved by the following refinement of schedule $\boldsymbol{h}_c^k$:

$$\boldsymbol{h}_c^k \leftarrow \boldsymbol{h}_c^k \setminus \left\{ u \in \boldsymbol{h}_c^k : \ \boldsymbol{N}^{k(u,c)}(u) \cap \boldsymbol{H}_c^{k(u,c)} \neq \emptyset \right\}.$$

In the example of Fig. A.1, we will seed user u_5 with content c_2 at time $k = 1$. If we know that $u_5 \in \boldsymbol{N}^3(u_1)$, i.e., that u_1 and u_5 will form a D2D connection at time $k = 3$ (i.e., when u_1 wants c_2) then then we can exclude u_1 from $\boldsymbol{h}_{c_2}^2$.

A.4 Dealing with Uncertainty

In Sec. A.3, we developed an optimal seeding strategy given the full and precise knowledge of the future (i) cellular background load, and (ii) predictable traffic pattern. Clearly, the performance of Proactive Seeding will strongly depend on the quality of our estimation of the predictable traffic $\boldsymbol{w}_c^k$. Many prediction techniques have been proposed in the literature and developing new ones is out of the scope of this chapter. Instead, in this section, we review some existing techniques, and we show how they can be incorporated in Proactive Seeding.

A.4.1 Interest diffusion on OSNs

In this chapter, we are interested in the content that becomes popular through social ties.[4] One can exploit the structure of the social network and information about interest diffusion, in order to predict information cascades. Such a prediction can then serve as input (instead of the offline knowledge) to our predictive seeding algorithms.

There is a rich literature on predicting the diffusion of interest in social networks, see e.g., [68, 69]. In our context, predicting the future progress of a cascade related to content c, can be modeled as finding the probability

$$\mathbb{P}\left(\boldsymbol{w}_c^{k+1},\boldsymbol{w}_c^{k+2},... \mid \boldsymbol{w}_c^k,\boldsymbol{w}_c^{k-1},...,\boldsymbol{w}_c^0, I_{other}\right), \tag{A.6}$$

[3] This information can be obtained either directly from the cellular network or can be contributed by the user e.g., via applications on OSNs (such as FourSquare, Facebook Places) or on a smartphone, in exchange for the service.

[4] An alternative approach to learn $\boldsymbol{w}_c^k$ could be by studying the download patterns of individual users. For example, assume that user u regularly visits `www.bbc.co.uk` (or checks out her Facebook updates) everyday in the morning. We can then seed u with some heavier content (graphics, videos) over night. A machine learning approach could help us choose whom to seed and with what content.

where $\boldsymbol{w}_c^k$,$\boldsymbol{w}_c^{k-1}$,...,$\boldsymbol{w}_c^0$ is the observed history at the current time k, and I_{other} represents any other available piece of information. Below, we comment on how some of the existing approaches translate into the (A.6) probabilities.

The threshold model

In the threshold model [68], each user u is associated with a threshold $0 \leq \theta_u \leq 1$. u becomes interested in the content at time $k+1$ if at least a (weighted) fraction of θ_u of her neighbors are interested in it at time k. This model is deterministic, i.e., the probabilities in (A.6) are either 0 or 1.

The cascade model

In the cascade model [68, 69], each edge (u,w) of the social graph is associated with an activation probability $q_{u,w}$. If user u gets interested in the content at time k, then the edge (u,w) is used exactly once to determine whether user w will become interested in the content at frame $k+1$, which happens with probability $q_{u,w}$. In other words, given the activation probabilities $q_{u,w}$ (i.e., I_{other}) and the history $\boldsymbol{w}_c^k$,$\boldsymbol{w}_c^{k-1}$,...,$\boldsymbol{w}_c^0$, the cascade model gives us the following probabilities, concerning the next time frame:

$$\mathbb{P}\left(\boldsymbol{w}_c^{k+1} \mid \boldsymbol{w}_c^k,\boldsymbol{w}_c^{k-1},...,\boldsymbol{w}_c^0,\, I_{other}\right), \tag{A.7}$$

which is a special case of (A.6).

Machine learning

Another line of research focuses on machine learning techniques that make use of all the available information. For example, in [70], the authors, based on the observed history, manage to accurately predict more than half of future re-tweets (of URL links) with 15% false positives.

A.4.2 From probabilities to Proactive Seeding

Given the knowledge of probabilities in (A.6), we follow the procedure presented in Fig. A.3. First, at the current time k, we use (A.6) to calculate the most likely future $\widehat{\boldsymbol{w}}_c^{l>k}$ (Fig. A.3(a)). Next, we plug $\widehat{\boldsymbol{w}}_c^{l>k}$ into Proactive Seeding (Fig. A.3(b)), which returns us the schedule $\boldsymbol{h}_c^k$ for the current time frame. Finally, we implement $\boldsymbol{h}_c^k$ and collect the actual evolution of demand $\boldsymbol{w}_c^k$ that is used to refine our calculations in the next time frame (Fig. A.3(c)). This means that our scheme is *adaptive* – at every iteration it updates the history by the current state of the network and recalculates $\boldsymbol{h}_c^k$.

Our prediction includes all times l between the current time k and time K. K is the latest time for which at least one realization of the interest diffusion process has at least

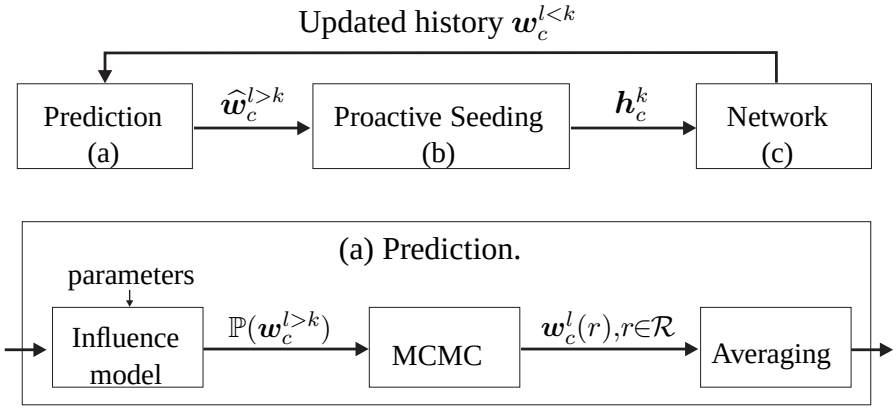


Figure A.3. Adaptive Proactive Seeding. (top) High-level overview. (bottom) The "Prediction" block.

one user interested in content c, i.e., $|\boldsymbol{w}_c^K| \geq 1$. For instance, for the cascade influence model, K is trivially upper-bounded by the total number of users, i.e., $K \leq |\boldsymbol{U}|$.

In Fig. A.3(bottom), we show in more detail the "Prediction" block from Fig. A.3. Given the knowledge of (A.6), we are, in principle, able to calculate exactly the expected future demand $\mathbb{E}[\boldsymbol{w}_c^{l>k}]$. In practice, however, the solution space is too big (especially if the number $|\boldsymbol{U}|$ of users or the final time K are large) to do it precisely. Instead, we run an MCMC (Monte Carlo Markov Chain) simulation, i.e., we use (A.6) to generate a number of realizations $\boldsymbol{w}_c^{l>k}(r)$, $r \in \mathcal{R}$. This step is illustrated by the middle block in Fig. A.3(bottom). Next, we average over all $|\mathcal{R}|$ realizations (right-most block in Fig. A.3, bottom), as follows.

First, we estimate the *number of users* $|\widehat{\boldsymbol{W}}_c^K|$ that eventually become interested the content, by the average over all the realizations:

$$|\widehat{\boldsymbol{W}}_c^K| = \frac{1}{|\mathcal{R}|} \cdot \sum_{r\in\mathcal{R}} |\boldsymbol{W}_c^K(r)|.$$

Next, we decide *which users* will become interested in the content, by taking $|\widehat{\boldsymbol{W}}_c^K|$ users with the highest observed probabilities $\widehat{\mathbb{P}}(u \in \boldsymbol{W}_c^K) = \frac{1}{|\mathcal{R}|} \cdot |\{r \in \mathcal{R} : u \in \boldsymbol{W}_c^K(r)\}|$ to request it. Finally, we interpret as $k(u,c)$ the time that is the most frequent across the realizations in R:

$$\widehat{k}(u,c) = \arg\max_{0\leq k\leq K} |\{r \in \mathcal{R} : u \in \boldsymbol{w}_c^k(r)\}|.$$

The above process provides an estimate $\widehat{\boldsymbol{w}}_c^k$ of the future demand, which we use as input to Proactive Seeding, as in Fig. A.3(b).

A.5 Evaluation

In this section, we evaluate the performance of Proactive Seeding through simulation.

A.5.1 Performance Metric

Without Proactive Seeding, user u fetches the content c over cellular when she wants it, which yields $\boldsymbol{h}_c^k \equiv \boldsymbol{w}_c^k$ and the peak cellular load equal to $\max_k \left(\lambda^k + \sum_c |\boldsymbol{w}_c^k|\right)$. In contrast, with Proactive Seeding, the peak cellular load drops to $\max_k \left(\lambda^k + \sum_c |\boldsymbol{h}_c^k|\right)$. Our main performance metric is the relative *gain* in peak cellular load, defined as

$$\gamma \;=\; \frac{\max_k \left(\lambda^k + \sum_c |\boldsymbol{w}_c^k|\right) - \max_k \left(\lambda^k + \sum_c |\boldsymbol{h}_c^k|\right)}{\max_k \left(\lambda^k + \sum_c |\boldsymbol{w}_c^k|\right)}.$$

Clearly, the larger the amount of the predictable traffic, the bigger gain γ we can expect. We therefore denote by ρ the ratio of the unpredictable traffic (aggregate over all contents) over the aggregate predictable traffic, i.e.,

$$\rho \;=\; \frac{\text{aggregated unpredictable traffic}}{\text{aggregated predictable traffic}} \;=\; \frac{\sum_k \lambda^k}{\sum_k \sum_c |\boldsymbol{w}_c^k|}. \tag{A.8}$$

A.5.2 Offline Scenario (using Twitter, Cellular and D2D traces)

First, we consider the offline case, with large-scale simulations fed by real traces of (a) interest diffusion process in Twitter [70], (b) background traffic from a US cellular operator [77], and (c) mobility [78]. This allows us to evaluate Proactive Seeding in presence of cellular background load and techniques that exploit D2D connectivity. We assume a priori knowledge of (a), (b), (c), and we evaluate how much gain γ is achieved by Proactive Seeding.

Description of Datasets

(a) Predictable traffic π^k: We use the Twitter trace from [70], where the authors collected the tweets that carry a URL (which defines our content), over a period of 300 hours (12.5 days). For our simulations, we kept only the "re-tweets" (indicated by an RT tag), which allows us to directly follow the cascades of interests in valuable (non-spam) content on Twitter (see also RT-cascades in [70]). Furthermore, in order to be able to observe the full evolution of such cascades, we exclude the URLs that appear in the first three or the last three hours of the trace. This leaves us with around 2.5M of tweets from 554K different users, sharing about 9000 contents (URLs). In Fig. A.4(a), we

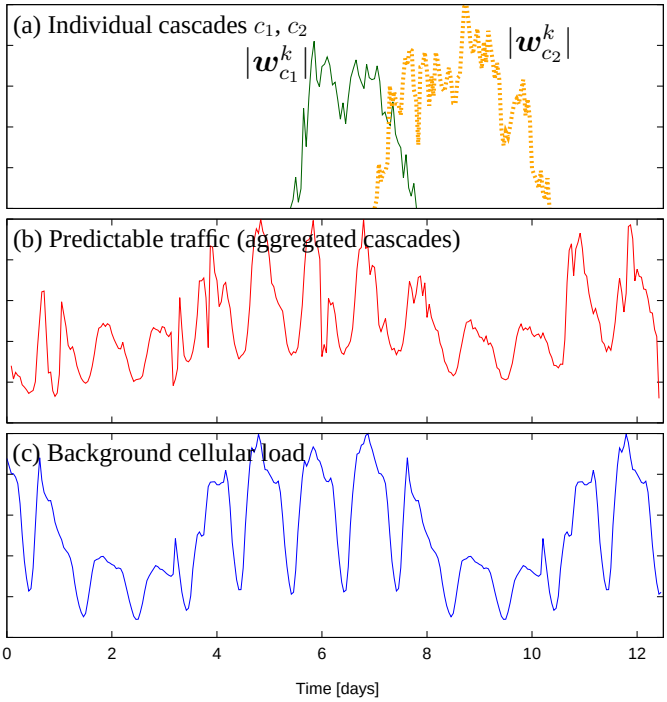


Figure A.4. Traces used in offline simulations. **(a)** Example of two individual Twitter cascades; **(b)** All 9000 Twitter cascades together [70]; **(c)** Background cellular load from a US operator [77]. For the sake of readability, all figures are normalized with respect to the peak value of the data they represent (i.e., they do not have the same scale).

show the evolution of two typical cascades from that trace. The "cascade" behavior is easy to see: the URL's popularity quickly increases over time, reaches a peak, and then declines. However, when we aggregate all the 9000 cascades together in Fig. A.4(b), the individual cascade shapes are not visible anymore; instead, the aggregated predictable traffic π^k clearly follows the daily pattern.[5]

(b) Background cellular load λ^k: As background load λ^k, we take a cellular traffic trace coming from a major operator in one US state [77].[6] Because this trace covers one full week (at a resolution of 1 hour), we replicate it, concatenate, and shift to match the 12.5 days of the Twitter trace. The result is presented in Fig. A.4(c). Similarly to Twitter, the cellular background load follows weekly and daily patterns.

(c) D2D connectivity: We use the Infocom06 contact trace [78] to simulate the device-to-device (D2D) connectivity. The trace logs the D2D contacts between 78 devices

[5] Recall, however, that our constraint (A.2) is defined for each content, not for the aggregated traffic.

[6] Strictly speaking, the trace [77] represents the total cellular traffic. For simplicity of presentation (e.g., independence of ρ), we interpret this trace as the background cellular load λ^k. We have also considered in simulations this trace as the total load, subtracting π^k to get the background load. The results in both cases are very similar.

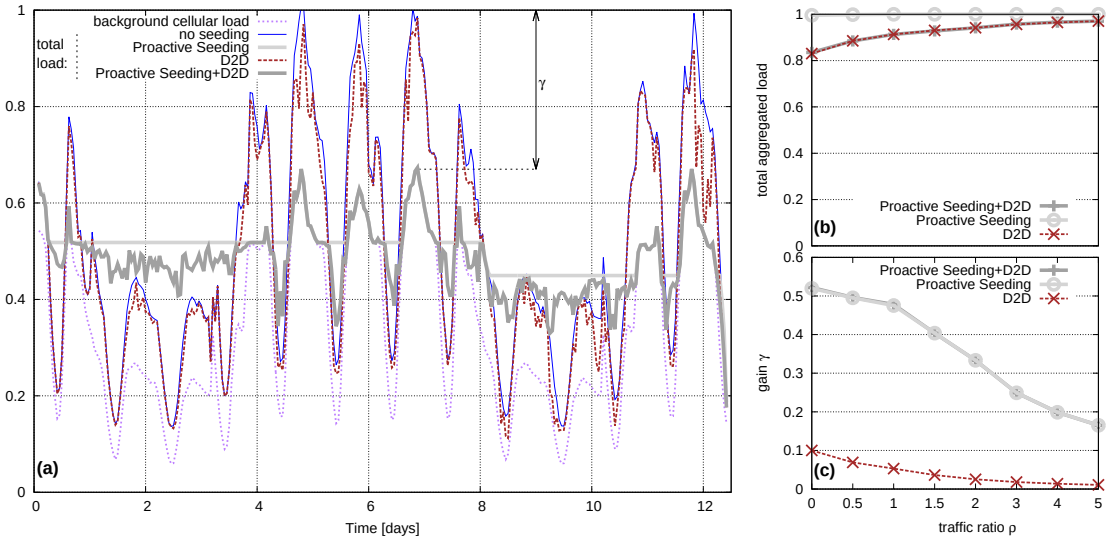


Figure A.5. Offline simulations driven by traces of (i) Twitter cascades (predictable traffic), (ii) background cellular load, and (iii) mobility. **(a)** Per hour time evolution of the total cellular load $\lambda^k + \pi^k$ under various scenarios, for traffic ratio of $\rho = 2$. **(b)** Aggregated cellular traffic as a function of ρ. **(c)** Gain γ as a function of ρ.

(iMotes) distributed to the attendees, over a period of three days.

For each content c, we randomly map the users $\boldsymbol{H}_c^K$ (i.e., eventually requesting c) to the users in the trace. Because of the limited size and duration of the trace, we replicate these users when $|\boldsymbol{H}_c^K| > 78$, and we repeat the connectivity pattern when the diffusion of interest in content c lasts for more than 3 days. Finally, users u and w are defined neighbors in our connectivity graph at hour k, i.e., $w \in \boldsymbol{N}^k(u)$ and $u \in \boldsymbol{N}^k(w)$, if u and w encounter each other within this hour (according to the Infocom06 trace).

The above mapping matches users $\boldsymbol{U}$ with nodes in the mobility trace in a purely random way. We also experimented with D2D connectivity graphs that reflect various levels of correlations between physical proximity and friendship. The results were similar and are omitted for lack of space.

Results

In Fig. A.5(a) we focus on a case when $\rho = 2$, i.e., the background load is twice the predictable traffic, and depict the time evolution of the total load on the 3G network in the following cases:

- no seeding: All users get the content they are interested in through the cellular network (i.e., $\boldsymbol{h}_c^k = \boldsymbol{w}_c^k, \forall c,k$).
- Proactive Seeding: Proactive Seeding algorithm is used to schedule predictable

traffic. D2D is disabled.

- D2D: Users exploit the D2D connectivity as explained in Sec. A.3.3, but Proactive Seeding is disabled.
- Proactive Seeding + D2D: predictable traffic is scheduled using Proactive Seeding *and* users exploit D2D links if available.

The no-seeding scenario results in a cellular load that is very uneven over time, with high peaks and periods of very low usage. Under D2D, we observe a slight reduction in the network load, with the peaks almost unchanged. In contrast, Proactive Seeding effectively reshapes the total cellular traffic, reducing the peaks by exploiting the less busy periods. Note that the peak load (around day 9) corresponds to a peak in the *background* load, which confirms that Proactive Seeding is optimal with respect to objective (A.4) (as we proved in Theorem A.3.1). Finally, when we combine Proactive Seeding and D2D, we observe a further reduction in the network load.

Fig. A.5(b) and Fig. A.5(c) show how the aggregated (i.e., over the whole trace duration) load and the gain γ depend on the ratio ρ between predictable and background load. Unsurprisingly, the higher ρ, the less beneficial Proactive Seeding becomes. Proactive Seeding effectively reduces the peak load (Fig. A.5(c)), but has no impact on the aggregated load (Fig. A.5(b)). The effect of D2D is quite the opposite. Applying both Proactive Seeding and D2D, we get the best of both worlds: i.e., a significant reduction in both the peak and the aggregated load.

A.5.3 The Online Case (using Diffusion Models on OSNs)

Sec. A.5.2 assumed full knowledge of the entire traces. In this section, we consider the case where the future can be predicted only with some amount of uncertainty, as described in Sec. A.4. For ease of explanation, we assume no background load and a single content c and we focus on evaluating the effect of uncertainty on the results.

Social Graphs (Datasets)

We use datasets from two different graphs, each capturing a different type of social tie.

- *Facebook*: The New Orleans network of the Facebook social graph [79], consisting of 63K vertices and 816K edges. The rationale for using this data set is that friends in Facebook share links and thus participate in spreading information about content.
- *Email*: a trace of e-mail contacts, consisting of 1133 nodes and 5452 edges. The rationale behind using this datasets is that emails often contain links that propagate in a viral way, leading to information cascades.

Social Influence (Models)

Using each of the previous graphs, we simulate interest diffusion through the cascade model [68, 69] described in Sec. A.4.1. We assume that 5% of users are interested in the content at time $k = 0$. The activation probability for each edge (u,w) is set to $q_{u,w} = 0.1$. (We have also tried a range of parameters, omitted for lack of space, and results were qualitatively similar.)

Uncertainty about the model and its parameters

Although the cascade model provides us with a probabilistic output, there are several other major sources of uncertainty about the future, which naturally lead to errors in the prediction. In particular, in practice, (i) we can never know exactly the model driving the spread of information and (ii) we can never know precisely the parameters of such a model. We capture these two effects in our simulations by introducing a multiplicative noise ν to the probabilities (A.6), i.e., we set $\mathbb{P}() \leftarrow \min(1,\nu\,\mathbb{P}())$. For example, $\nu = 1.2$ results in a systematic overestimation of the future demand by 20%, and $\nu = 0.8$ underestimates it by 20%.

Results

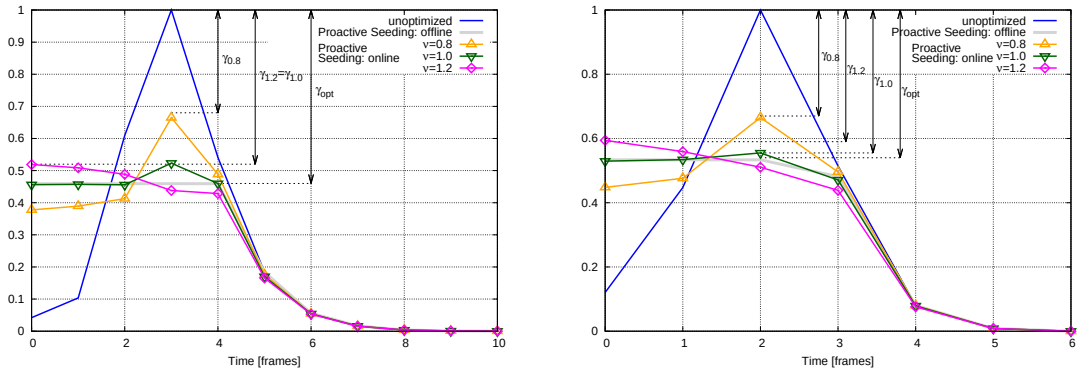


Figure A.6. Online simulations on the Facebook (left) and Email (right) graphs.

In Fig. A.6, we present results for the Facebook (left) and Email (right) graphs. Although the two networks are very different in size and structure, they exhibit the same qualitative behavior, with a clear cascade evolution. The way Proactive Seeding works is easy to observe: the users known (or assumed) to request the content during the peak time are served during earlier frames, thus reducing the peak load.

For both networks, we compare the ideal (i.e., offline) performance with the adaptive (i.e., online) case, in which the demand is not known a priori. In the latter, we consider three values of the noise ν. If our prediction is not systematically biased ($\nu = 1$), the online performance of Proactive Seeding is close to the optimal (offline). In contrast, systematically overestimating ($\nu > 1$) or underestimating ($\nu < 1$) the future demand leads to less gain γ, but with qualitatively different effects. *Overestimating* the demand means serving users that will never need the content, thus wasting network and user resources. In the extreme case, it may even lead to a negative gain, i.e., a peak load $\max_k |\boldsymbol{h}_c^k|$ greater than the peak demand $\max_k |\boldsymbol{w}_c^k|$. On the other hand, *underestimating* the demand is conservative, as moves towards the no-seeding case. The gain γ can decrease, but is still above zero. Therefore, as a practical take-away from our online evaluation, we can recommend to tune the prediction parameters so as to underestimate rather than overestimation the demand.

Fig. A.6 also allows us to see how the adaptiveness, i.e., the fact that at each time frame k we feed the actual set $\boldsymbol{W}_c^k$ of users interested in the content back to the prediction algorithm, allows us to recover from prediction errors. If $\nu > 1$, we tend to overestimate the number of users interested in the content at the begin of the cascade. However, as we observe the actual number of interested users, we are able to correct the error, and schedule fewer users in the subsequent frames. Conversely, if $\nu < 1$, we start seeding fewer users than we should, and we make it up for this error later. Notice however that both such cases imply a peak load that is higher than the ideal (i.e., offline) one.

A.6 Related work

Proactive Seeding touches upon several research areas. We now review the closest ones and how they relate to our work.

Opportunistic communication. When several users are interested in the same content and they are in proximity of each other, some of them may be able to use device-to-device connections, e.g., through WiFi or Bluetooth, to get the content, instead of their cellular connection. This opportunistic communication results in offloading the cellular network. In [64], device-to-device and cellular connections are used to disseminate dynamic content, so as to maximize the "freshness"' of the content. The connectivity of nodes are taken into account in order to select the right users to act as relays. As an example, a node with many neighbors is more likely to be selected as a relay. The work in [74] considers a similar scenario and assumes that social ties among the users are strongly correlated with their physical proximity and similar interests. [65] offloads the cellular network through proximity connections, while still meeting strict deadlines. With respect to these works, we have a different goal – decreasing the *peak* load on the cellular network – and a stronger constraint, i.e., the user impatience.

Socially-aware forwarding. Another body of work [55, 80–84] exploits the principle that social ties affect the mobility, and eventually the proximity, of users. Evidence has been provided, for example in [76], which shows that there is a significant correlation between similar interests and geographical proximity, for four different OSNs (BrightKite, FourSquare, LiveJournal and Twitter). Therefore, knowledge about social ties, can be taken into account to optimize routing for content delivery.

[80] presents Bubble rap – a routing protocol for DTNs. Devices detect the centrality of the community the user belongs to, based on the frequency of contact. This is then used for routing decisions. [81, 82] use social information to optimize content discovery in a publish/subscribe setting: the more social users are given a special role in the delivery process. [55, 83, 84] exploit social information to route queries and to decide which items should be cached or duplicated.

In our work, we exploit social ties for a different purpose, namely predicting the content requests in order to proactively serve them. Furthermore, we limit the amount of information that users disclose to their peers (e.g., users do not broadcast their whole list of topics of interest, as in [55]).

Interest diffusion in social networks. There is a large body of literature on diffusion in networks, including but not limited to technological networks. The classic work in [85] reviews several influence models and proposes an algorithm for selecting which nodes to seed so as to maximize the diffusion, given the social structure. This is different from our objective in this chapter (to minimize the peak of the cascade) as well as in the fact that seeding is done only once in the beginning, while we adaptively seed at every time slot.

Such influence models are motivated by the many studies of information diffusion on actual social networks. For example, [30] identifies and studies several cascades on the Flickr social network. [86] analyzes 1.5 million YouTube videos, showing that not all popular videos are "social" and that highly social videos rise to, and fall from, their peak popularity more quickly than less social videos. Somewhat related to our work, [63] considers information cascades caused by social influence and shows which links to select and limit this influence, so as to delay the peak of the load caused on the cellular network.

Predicting content popularity. Forecasting the popularity of content, with or without taking into account network effects, is another active research area. [87] presents methods for predicting the popularity of items given historical access data, but without taking into account the network effect, for the YouTube and Digg social networks. [70] collects a dataset of 22M tweets, containing 15M URLs and presents a methodology (based on influence models) which predicts more than half of the tweets in the dataset with only 15% false positives.

In this chapter, we use the dataset collected in [70] for simulations of the offline scenario. More generally, we rely on prediction models as a part of our machinery, but

we do not develop one ourselves.

A.7 Conclusion

We presented proactive seeding for information cascades in social media - as a new technique to reduce the peak demand in cellular networks. In the special case of single content with no background load, the optimal solution that minimizes the peak load turns out to have an intuitive interpretation. In the general case of multiple contents with known background traffic, we provide a greedy algorithm and prove its optimality, in the offline case. In the online case, we investigated the performance of the proposed solutions by replacing the actual future demand by the predicted demand. Our evaluation showed robustness, especially when underestimating the total demand. We also extended our algorithm to take into account D2D communication, when this is available, thus offloading the total cellular traffic, in addition to reducing the peak load. Our evaluation over real traces indicate that proactive seeding via predicting social cascades significantly reduces the peak load as much as 50%.

Acknowledgements

We would like to thank the authors of [70] and [77] , for providing the Twitter and 3G traffic traces, respectively.

Bibliography

[1] Z. Zheng, P. Sinha, S. Kumar, "Alpha coverage: bounding the interconnection gap for vehicular Internet access," *IEEE INFOCOM,* Rio de Janeiro, Brazil, Apr. 2009.

[2] Z. Zheng, Z. Lu, P. Sinha, S. Kumar, "Maximizing the contact opportunity for vehicular Internet access," *IEEE INFOCOM,* San Diego, CA, Mar. 2010.

[3] M. Fiore, J. M. Barcelo-Ordinas, "Cooperative download in urban vehicular networks," *IEEE MASS,* Macau, China, Oct. 2009.

[4] D. Hadaller, S. Keshav, T. Brecht, S. Agarwal, "Vehicular opportunistic communication under the microscope," *ACM MobySys,* San Juan, Puerto Rico, June 2007.

[5] B. B. Chen, M. C. Chan, "MobTorrent: A framework for mobile Internet access from vehicles," *IEEE INFOCOM,* Rio de Janeiro, Brasil, Apr. 2009.

[6] J. Zhao, T. Arnold, Y. Zhang, G. Cao "Extending drive-thru data access by vehicle-to-vehicle relay," *ACM VANET,* San Francisco, CA, Sept. 2008.

[7] J. Ott, D. Kutscher, "Drive-thru Internet: IEEE 802.11b for "Automobile" users," *IEEE INFOCOM,* Hong Kong, China, Mar. 2004.

[8] V. Bychkovsky, B. Hull, A. K. Miu, H. Balakrishnan, S. Madden, "A measurement study of vehicular Internet access using in situ Wi-Fi networks," *ACM MobiCom,* Los Angeles, CA, Sept. 2006.

[9] G. Marfia, G. Pau, E. Giordano, E. De Sena, M. Gerla, "Evaluating vehicle network strategies for downtown Portland: opportunistic infrastructure and importance of realistic mobility models," *ACM MobiOpp,* San Juan, Puerto Rico, June 2007.

[10] Y. Ding, C. Wang, L. Xiao, "A static-node assisted adaptive routing protocol in vehicular networks," *ACM VANET,* Montreal, Canada, Sept. 2007.

[11] C. Lochert, B. Scheuermann, C. Wewetzer, A. Luebke, M. Mauve, "Data aggregation and roadside unit placement for a VANET traffic information system," *ACM VANET,* San Francisco, CA, Sept. 2008.

[12] O. Trullols, M. Fiore, C. Casetti, C.-F. Chiasserini, J. M. Barcelo-Ordinas, "Planning roadside infrastructure for information dissemination in intelligent transportation systems," *Computer Comm.,* vol. 33, no. 4, pp. 432–442, Mar. 2010.

[13] A. Abdrabou, W. Zhuang, "Probabilistic delay control and road side unit placement for vehicular ad hoc networks with disrupted connectivity," *IEEE JSAC,*

vol. 29, no. 1, Jan. 2011.

[14] V. Kone, H. Zheng, A. Rowstron, B.Y. Zhao, "On infostation density of vehicular networks," *Annual International Wireless Internet Conference (WICON)*, Singapore, Mar. 2010.

[15] A. Balasubramanian, B.N. Levine, A. Venkataramani, "Enhancing interactive Web applications in hybrid networks," *ACM/IEEE MobiCom*, San Francisco, CA, Sept. 2008.

[16] S. Yoon, D. T. Ha, H. Q. Ngo, C. Qiao, "MoPADS: A mobility profile aided file downloading service in vehicular networks," *IEEE Trans. on Veh. Tech.*, vol. 58, no. 9, Nov. 2009.

[17] A. Nandan, S. Das, G. Pau, M. Gerla, M. Y. Sanadidi, "Co-operative downloading in vehicular ad-hoc wireless networks," *IEEE WONS*, St.-Moritz, Switzerland, Jan. 2005.

[18] M. Gerla, M. Gruteser, "Vehicular networks: Applications, protocols, and testbeds," in *Emerging Wireless Technologies and the Future Mobile Internet*, Eds.: D. Raychaudhuri, M. Gerla, May 2011.

[19] M. Sardari, F. Hendessi, F. Fekri, "Infocast: A new paradigm for collaborative content distribution from roadside units to vehicular networks," *IEEE SECON*, Rome, Italy, June 2009.

[20] O. Trullols-Cruces, J. Morillo, J. M. Barcelo-Ordinas, J. Garcia-Vidal, "A cooperative vehicular network framework," *IEEE ICC*, Dresden, Germany, June 2009.

[21] N. Banerjee, M.D. Corner, D. Towsley, B.N. Levine, "Relays, meshes, base stations: Enhancing mobile networks with infrastructure," *ACM/IEEE MobiCom*, San Francisco, CA, Sept. 2008.

[22] D. Hay, P. Giaccone, "Optimal routing and scheduling for deterministic delay tolerant networks," *IEEE WONS*, Snowbird, UT, Feb. 2009.

[23] F. Malandrino, C. Casetti, C.-F. Chiasserini, M. Fiore, "Content downloading in vehicular networks: what really matters," *IEEE INFOCOM Mini-Conference*, Shanghai, China, Apr. 2011.

[24] G. Iosifidis, I. Koutsopoulos, G. Smaragdakis, "The impact of storage capacity on end-to-end delay in time varying networks," *IEEE INFOCOM*, Shanghai, China, Apr. 2011.

[25] L. R. Ford, D. R. Fulkerson, *Flows in networks*, Princeton Univer- sity Press, Princeton, NJ, 1962.

[26] L. Lovasz, "Random walks on graphs. a survey," *Combinatorics*, 1993.

[27] N. Cetin, A. Burri, K. Nagel, "A large-scale multi-agent traffic microsimulation based on queue model," *STRC*, Ascona, Switzerland, Mar. 2003.

[28] F. Aidouni, M. Latapy, C. Magnien, "Ten weeks in the life of an eDonkey server," *International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P)*, Rome, Italy, 2009.

[29] M. Lacage, M. Hossein Manshaei, T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," *ACM MSWiM*, Venice, Italy, Oct. 2004.

[30] M. Cha, A. Mislove, K. P. Gummadi, "A measurement-driven analysis of information propagation in the Flickr social network," *WWW*, 2009.

[31] Y. Wu, Y. Zhu, B. Li, "Trajectory improves data delivery in vehicular networks," *Infocom*, 2011.

[32] H. Zhu, S. Chang, M. Li, S. Naik, S. Shen, "Exploiting temporal dependency for opportunistic forwarding in urban vehicular networks," *Infocom*, 2011.

[33] TomTom, "How TomTom's HD TrafficTM and IQ RoutesTM data provides the very best routing," *White chapter*, 2010.

[34] F. Malandrino, C. Casetti, C.-F. Chiasserini, M. Fiore, "Representing a Markovian prediction technique through a fog-of-war model," `http://www.telematica.polito.it/malandrino/techrep-markov.pdf`, 2011.

[35] C. Barberis, G. Malnati, "Epidemic information diffusion in realistic vehicular network mobility scenarios," *ICUMT*, 2009.

[36] E. Cohen, S. Shenker, "Replication strategies in unstructured peer-to-peer networks," *Sigcomm*, 2002.

[37] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, M. Dias de Amorim, "Relieving the wireless infrastructure: When opportunistic networks meet guaranteed delays," *WoWMoM*, 2011.

[38] J. Ahn, B. Krishnamachari, F. Bai, L. Zhang, "Optimizing content dissemination in heterogeneous vehicular networks," *USC CENG Tech. Rep. CENG-2010-2*, 2010.

[39] R. L., Cruz, A. V. Santhanam, "Optimal routing, link scheduling and power control in multihop wireless networks," *Infocom*, 2003.

[40] W. Gao, G. Cao, "User-centric data dissemination in disruption tolerant networks," *Infocom*, 2011.

[41] U. G. Acer, P. Giaccone, D. Hay, G. Neglia, S. Tarapiah, "Timely data delivery in a realistic bus network," *Infocom Mini-Conference*, 2011.

[42] B. B. Chen, M. C. Chan, "MobTorrent: A framework for mobile Internet access from vehicles," *Infocom*, 2009.

[43] X. Li, X. Yu, A. Wagh, C. Qiao, "Human factors-aware service scheduling in vehicular cyber-physical systems," *Infocom*, 2011.

[44] "YANS, Yet Another Network Simulator," INRIA Research Report, available at http://hal.inria.fr/inria-00078318/en/

[45] L. Buttyan, J.-P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 8, pp. 579-592, 2003.

[46] L. Liquori, D. Borsetti, C. Casetti, C.-F. Chiasserini, "An Overlay Architecture for Vehicular Networks," *Networking*, 2008.

[47] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, R. Rao, "An Analytical Approach

to the Study of Cooperation in Wireless Ad Hoc Networks," *IEEE Trans. on wireless and communications*, vol. 4, no. 2, 2005.

[48] N. Salem, L. Buttyan, J. Hubaux, M. Jakobsson, "Node Cooperation in Hybrid Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 4, April 2006.

[49] Y. Zhang, W. Lou, Y. Fang, "A Secure Incentive Protocol for Mobile Ad Hoc Networks," *ACM Wireless Networks*, vol. 13, no. 5, pp. 569–582, October 2007.

[50] M. E. Mahmoud, X. Shen, "Stimulating Cooperation in Multi-hop Wireless Networks Using Cheating Detection System," *IEEE INFOCOM*, 2010.

[51] P. Michiardi, R. Molva, "Core: a Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks, *Conference on Communications and Multimedia Security*, 2002.

[52] J. Jaramillo, R. Srikant, "DARWIN: Distributed and Adaptive Reputation Mechanism for Wireless Networks," *ACM MobiCom*, 2007.

[53] L. Xie, S. Zhu, "Message Dropping Attacks in Overlay Networks: Attack Detection and Attacker Identification," *ACM Trans. on Information and System Security*, vol. 11, no. 3, March 2008.

[54] M. X. Goemans, *et al.*, "Market Sharing Games Applied to Content Distribution in Ad-Hoc Networks," *ACM MobiHoc*, 2004.

[55] C. Boldrini, M. Conti, A. Passarella, "ContentPlace: Social-aware Data Dissemination in Opportunistic Networks," *IEEE/ACM MSWiM*, 2008.

[56] F. R. Dogar, A. Phanishayee, H. Pucha, O. Ruwase, D. G. Andersen, "Ditto: a System for Opportunistic Caching in Multi-hop Wireless Networks," *ACM MobiCom*, 2008.

[57] W. W. Terpstra, J. Kangasharju, C. Leng, A. P. Buchmann, "BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search," *ACM SIGCOMM*, 2007.

[58] F. Malandrino, C. Casetti, C.-F. Chiasserini, "Content Discovery and Caching in Mobile Networks," *IEEE PIMRC* , Istanbul, Turkey, Sept. 2010.

[59] M. Felegyhazi, J.-P. Hubaux, "Game Theory in Wireless Networks: A Tutorial," LCA-REPORT-2006-002, 2006.

[60] M. D. Lillibridge, M. Adabi, K. Bharat, A. Broder, "Method for Selectively Restricting Access to Computer Systems," *Tech. Rep., US Patent 6,195,698*, 2001.

[61] L. A. Wolsey, *Integer Programming*, Wiley, San Francisco, 1998.

[62] Facebook, "Facebook statistics," http://www.facebook.com/press/info.php?statistics, July 2011.

[63] H. Sharara, C. Westphal, S. Radosavac, and U. C. Kozat, "Utilizing Social Influence in Content Distribution Networks," *IEEE ICC*, 2011.

[64] S. Ioannidis, A. Chaintreau, and L. Massoulie, "Optimal and Scalable Distribution of Content Updates over a Mobile Social Network," *IEEE INFOCOM*, Apr. 2009.

[65] J. Whitbeck, Y. Lopez, J. Leguay, V. Conan, and M. De Amorim, "Relieving the

wireless infrastructure: When opportunistic networks meet guaranteed delays," *Arxiv preprint arXiv:1007.5459*, 2010.

[66] Credit Suisse, "U.S. wireless networks running at 80% of capacity," http://benton.org/node/81874, July 2011.

[67] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015," *Cisco white chapter*, 2011.

[68] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *ACM SIGKDD*, 2003.

[69] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.

[70] W. Galuba, K. Aberer, D. Chakraborty, Z. Despotovic, and W. Kellerer, "Outtweeting the twitterers-predicting information cascades in microblogs," in *WOSN*, 2010.

[71] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Trans. on Mobile Computing*, vol. 5, no. 1, pp. 77–89, Jan. 2006.

[72] H. Gupta and S. Das, "Benefit-Based Data Caching in Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, vol. 7, no. 3, pp. 289–304, Mar. 2008.

[73] 3GPP Work Item Description TSG-RAN, "Study on LTE device to device discovery and communication," RP-110707 LTE-D2D RAN, May 2011.

[74] B. Han, P. Hui, V. Kumar, M. Marathe, G. Pei, and A. Srinivasan, "Cellular traffic offloading through opportunistic communications: a case study," in *ACM CHANTS*. ACM, 2010.

[75] Qualcomm, "Flashlinq," http://www.qualcomm.com/stories/2011/02/08/ phone-conversations-minus-people, February 2011.

[76] S. Scellato, C. Mascolo, M. Musolesi, and V. Latora, "Distance matters: Geo-social metrics for online social networks," in *WOSN*, 2010.

[77] M. Shafiq, L. Ji, and A. Liu, "Characterizing and modeling internet traffic dynamics of cellular devices," *ACM SIGMETRICS*, 2011.

[78] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "CRAWDAD data set cambridge/haggle (v. 2009-05-29)," Downloaded from http://crawdad.cs.dartmouth.edu/cambridge/haggle, May 2009.

[79] B. Viswanath, A. Mislove, M. Cha, and K. Gummadi, "On the evolution of user interaction in facebook," in *WOSN*, Barcelona, Spain, 2009.

[80] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: social-based forwarding in delay tolerant networks," *IEEE Trans. on Mobile Computing*, 2010.

[81] E. Yoneki, P. Hui, S. Chan, and J. Crowcroft, "A socio-aware overlay for publish/subscribe communication in delay tolerant networks," in *ACM MSWiM*, 2007.

[82] P. Costa, C. Mascolo, M. Musolesi, and G. Picco, "Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks," *IEEE JSAC*, vol. 26,

no. 5, pp. 748–760, Jun. 2008.

[83] E. Jaho and I. Stavrakakis, "Joint interest- and locality-aware content dissemination in social networks," in *IEEE/IFIP WONS*, Feb. 2009.

[84] A. J. Mashhadi, S. B. Mokhtar, and L. Capra, "Habit: Leveraging human mobility and social network for efficient content dissemination in Delay Tolerant Networks," in *IEEE WoWMoM*, Jun. 2009.

[85] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[86] T. Broxton, Y. Interian, J. Vaver, and M. Wattenhofer, "Catching a Viral Video," in *IEEE ICDM Workshops*, 2010.

[87] G. Szabó and B. Huberman, "Predicting the popularity of online content," *Communications of the ACM*, vol. 53, no. 8, pp. 80–88, Aug. 2008.